Direct and Indirect Multimodal Few-Shot Learning of Speech and Images

by

Leanne Nortje

Pectora roborant cultus recti

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University.

Supervisor: Dr H. Kamper
Department of Electrical and Electronic Engineering

December 2020

Stellenbosch University https://scholar.sun.ac.za

Acknowledgements

- ≫ My supervisor Dr Herman Kamper, I have so much gratitude for the support, patience and encouragement you gave in the past two years. I am grateful to have learnt so much from you.
- ≫ I would firstly like to thank the CSIR DST-scholarship and Saigen for providing the funding to complete my studies.
- ≫ To the LSL-research group and the friends in the MediaLab, thanks for the lego building sessions, the funny videos and the coffee breaks which turned into motivational talks. Without this life in the lab would have been dull.
- ≫ Jo-Anelda, thanks for your unwavering support this past fourteen years. Thanks for just listening to my rants at our weekly dinners.
- ≫ Claudia, I really appreciated our coffee breaks and the advice that normally came with it. Thanks for making sure that I took breaks and got out a bit when the work piled up.
- ≫ I would like to thank my parents and sister for their support and the opportunity to be able to study. To my sister Es-Marié and brother(-in-law) Hannes, thanks for the interest and emotional support in the form of puppy videos and pictures.
- ≫ To James and Mary, your video calls always brightened my day. Thank you for all the times I could break away and come visit you.
- ≫ I am eternally grateful to the Lord for the opportunities bestowed on me and the strength to conquer this amazing challenge.

“ *I lift up my eyes to the mountains–*
where does my help come from?
My help comes from the Lord,
the Maker of heaven and earth. ”
– Psalm 121:1-3

Stellenbosch University https://scholar.sun.ac.za

Declaration

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.
2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.
3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.
4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism
5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Leanne Nortje **Voorletters en van / *Initials and surname***	4 September 2020 **Datum / *Date***

Copyright ©2020 Stellenbosch University
All rights reserved.

Stellenbosch University https://scholar.sun.ac.za

Abstract

Children have the ability to learn new words and corresponding visual objects from only a few word-object example pairs. This raises the question of whether we can find multimodal speech-vision systems which can learn as rapidly from only a few example pairs. Imagine an agent like a household robot is shown an image along with a spoken word describing the object in the image, e.g. *teddy*, *monkey* and *dog*. After observing a single paired example per class, it is shown a new set of unseen pictures, and asked to pick the "teddy". This problem is referred to as *multimodal one-shot matching*. If more than one paired speech-image example is given per concept type, it is called *multimodal few-shot matching*. In both cases, the set of initial paired examples is referred to as the *support set*.

This thesis makes two core contributions. Firstly, we compare unsupervised learning to transfer learning for an indirect multimodal few-shot matching approach on a dataset of paired isolated spoken and visual digits. Transfer learning (which was used in a previous study) involves training models on labelled background data not containing any of the few-shot classes; it is conceivable that children use previously gained knowledge to learn new concepts. It is also conceivable that prior to seeing the few-shot pairs, a household robot or child would be exposed to unlabelled in-domain data from its environment; we therefore consider unsupervised learning for this problem which we are also the first to do. In unsupervised learning, models are trained on unlabelled in-domain data. From all our experiments, we find that transfer learning outperforms unsupervised learning.

Indirect models (which were used in our first contribution) consist of two separate unimodal networks with the support set acting as a pivot between the modalities. In contrast, a direct model would learn a single multimodal space in which representations from the two modalities can be directly compared. We propose two new direct multimodal networks: a multimodal triplet network (MTriplet) which combines two triplet losses, and a multimodal correspondence autoencoder (MCAE) which combines two correspondence autoencoders (CAEs). Both these models require paired speech-image examples for training. Since the support set is not sufficient for this purpose, we propose a new pair mining approach in which pairs are constructed automatically from unlabelled in-domain data using

Stellenbosch University https://scholar.sun.ac.za

the support set as a pivot. This pair mining approach combines unsupervised and transfer learning, since we use transfer learned unimodal classifiers to extract representations for the unlabelled in-domain data. We show that these direct models consistently outperform the indirect models, with the MTriplet as the top performer. These direct few-shot models show potential towards finding systems that learn from little labelled data while being capable of rapidly connecting data from different modalities.

Stellenbosch University https://scholar.sun.ac.za

Uittreksel

Kinders het die vermoë om nuwe woorde en ooreenstemmende visuele voorwerpe te leer van slegs ‘n paar oudiovisuele voorbeeldpare. Dit bring die vraag na vore of ons veelvuldige-modaliteit oudiovisuele sisteme kan kry wat so vinnig van ‘n paar voorbeeldpare kan leer. Stel jou voor dat daar vir ‘n agent soos ‘n huishoudelike robot, ‘n beeld met ‘n gesproke woord wat die voorwerp in die beeld beskryf, gegee word, b.v. *teddiebeer*, *apie* en *hond*. Nadat ‘n enkele voorbeeld paar per klas waargeneem is, word die agent gevra om die "teddiebeer" in ‘n nuwe stel beelde te kies. Daar word na die probleem verwys as *veelvuldige-modaliteit eenskoot-passing*. Indien meer as een oudiovisuele voorbeeld paar gegee is vir elke konsep tipe, word dit *veelvuldige-modaliteit meerskoot-passing* genoem. In beide gevalle verwys ons na die stel oorspronklike voorbeeldpare as die *ondersteuningsstel*.

Hierdie proefskrif maak twee kern bydraes. Eerstens, vergelyk ons sonder-toesig-leer teenoor oordragsleer vir ‘n indirekte veelvuldige-modaliteit meerskoort-passing benadering op ‘n datastel van ooreenstemmende beelde en geïsoleerde gesproke syfers. Oordragsleer (wat in ‘n vorige studie gebruik is) behels die afrig van modelle op agtergrond data wat nie enige van die meerskoot klasse bevat nie; dit word gemotiveer aangesien kinders kennis gebruik wat hulle voorheen opgedoen het om nuwe konsepte te leer. Voor die huishoudelike robot of kind die meerskoot pare sien, is dit ook moontlik dat hy/sy vanaf die omgewing blootgestel word aan binne-domein data sonder annotasies. Ons oorweeg daarom leer-sonder-toesig vir die probleem en is die eerstes om dit te doen. In leer-sonder-toesig, word modelle afgerig op binne-domein data sonder annotasies. Gebasseer op al ons eksperimente, vind ons dat oordragsleer beter as leer-sonder-toesig presteer.

Indirekte modelle (wat in ons eerste bydrae gebruik is) bestaan uit twee aparte enkelmodaliteit netwerke met die ondersteuningsstel wat dien as ‘n spilpunt tussen die modaliteite. In plaas hiervan leer ‘n direkte model ‘n enkele veelvuldige-modaliteit-ruimte waarin voorstellings vanaf twee modaliteite direk vergelyk kan word. Ons stel twee nuwe direkte modelle voor: ‘n veelvuldige-modaliteit drieling-model (VMDrieling) wat twee drieling koste-funksies kombineer, en ‘n veelvuldige-modaliteit korrespondensie-outo-enkodeerder (VMOE) wat twee outo-enkodeerders (OEs) kombineer. Al die modelle vereis

Stellenbosch University https://scholar.sun.ac.za

gepaarde oudiovisuele voorbeelde tydens afrigting. Aangesien die ondersteuningsstel nie voldoende is hiervoor nie, stel ons ‘n nuwe ontginningsskema voor waarin pare automaties opgestel word vanaf binne-domein data sonder annotasies, met die ondersteuningstel wat as ‘n spilpunt gebruik word. Hierdie ontginningsskema kombineer oordragsleer en leer-sonder-toesig aangesien ons enkelmodaliteit klassifiseerders wat afgerig is met oordragsleer gebruik om voorstellings vir binne-domein data sonder annotasies, te verkry. Ons wys dat hierdie direkte modelle konsekwent beter presteer as die indirekte modelle, met die VMDrieling as die beste presteerder. Hierdie direkte modelle toon potensiaal om sisteme te vind wat van min geannoteerde data leer terwyl dit terselfdertyd data vanaf verskillende modaliteite aanmekaar kan verbind.

Stellenbosch University https://scholar.sun.ac.za

Table of Contents

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

List of Figures

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

List of Tables

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Nomenclature

Functions

ℓ	A loss objective function.
σ	An activation function.
$C_{\mathcal{S}}$	A classification metric.
$D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$	A distance metric between speech instances $\mathbf{x}_a$ and image instances $\mathbf{x}_v$.
$f_{\boldsymbol{\Theta}}(\mathbf{x})$	A model function.
$g_{\boldsymbol{\Phi}}(\mathbf{x})$	The function of a FFNN, CNN or RNN.

Variables

α	A weighting parameter.
β	The learning rate.
$\mathbf{b}$	A bias vector.
$\mathbf{u}$	The input vector to a layer.
$\mathbf{v}$	The layer output.
$\mathbf{W}$	A weight matrix.
$\mathbf{x}$	Some input matrix or vector.
$\mathbf{X}^{(i)}$	All the model inputs for a single training example.
$\mathbf{x}_a^*$	A speech query.
$\mathbf{x}_a^{(i)}$	A spoken word instance.
$\mathbf{x}_v^{(i)}$, $\mathbf{x}_v^{(j)}$	The i^{th} or j^{th} image instance.
$\mathbf{x}_{\text{neg}}$	A negative to $\mathbf{x}$.

Stellenbosch University https://scholar.sun.ac.za

$\mathbf{x}_{\text{pair}}$	A pair of $\mathbf{x}$.
$\mathbf{y}$	The desired output of a network.
$\mathbf{Y}^{(i)}$	All the model's desired outputs for a single training example.
$\mathbf{z}^{(i)}$	A feature representation.
$\mathbf{z}_a$	The feature representation for a given speech instance.
$\mathbf{z}_v$	The feature representation of a given image instance.
$\mathbf{\Phi}$	The trainable parameters of a FFNN, CNN or RNN.
$\boldsymbol{\phi}$, $\boldsymbol{\theta}$	The trainable parameters of a subnetwork.
$\mathbf{\Theta}$	The trainable parameters of an entire model.
η	The number of frames in a word sequence.
$\hat{\mathbf{y}}$	The output vector of a network.
$\hat{\mathbf{Y}}^{(i)}$	All the model outputs for a single training example.
$\mathcal{B}$	A set of batch size values.
$\mathcal{M}_v$	A matching set of images.
$\mathcal{S}$	A uimodal or multimodal support set.
$\mathrm{y}_a^{(i)}$	The word label of a spoken word.
τ	The number of training examples in a batch.
K	The given number of examples per class in a support set.
k	The amount of sampled examples per class.
L	The number of classes.
M	The dimension of the layer before the layer with dimension N.
m	A margin parameter.
N	The dimension of a layer.
p	The amount of sampling classes.

Stellenbosch University https://scholar.sun.ac.za

Acronyms and Abbreviations

AE	Autoencoder
CAE	Correspondence Autoencoder
CNN	Convolutional Neural Network
DL	Deep Learning
DTW	Dynamic Time Warping
FFNN	Feedforward Neural Network
GRU	Gated Recurrent Unit
HBPL	Hierarchical Bayesian Program Learning
HHMM	Hierarchichal Hidden Markov Model
LSTM	Long Short-term Memory
MAML	Model-agnostic Meta-learning
MCAE	Multimodal Correspondence Autoencoder
MFCC	Mel-frequency Cepstral Coefficient
ML	Machine Learning
MTriplet	Multimodal Triplet Network
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Decent
VAE	Variational Autoencoder

Stellenbosch University https://scholar.sun.ac.za

Chapter 1

Introduction

In the past couple of years deep learning (DL) has led to substantial improvements in speech and image recognition systems. However, since DL methods are typically very data dependant, this causes the ripple effect of current audio and vision recognition systems requiring large amounts of transcribed speech and labelled image data. Since transcribing and labelling data is expensive and time-consuming [2], this data dependency has led to numerous research studies into *one-shot learning* [3–9] to find machine learning (ML) solutions that use less transcriptions and labels. One-shot learning is a problem formulation in which a model needs to learn a new concept from only *one* labelled example. One-shot learning can be extended to *few-shot* learning in which a model learns a new concept from a *few* labelled examples of the concept instead of just one. For example, in the one-shot setting, imagine a model is presented with spoken words and their corresponding word labels, e.g "cake", "cookie", "milk" and "juice". After hearing these words with their corresponding labels only once, the model is presented with another instance of the word "cake" from which it should identify the cake label.

Humans are able to learn new words and objects in a *one-shot* (once-off) manner [10]. For example, after a child hears a novel word once, the child can infer the likely meaning of the word [11]. However, young children do not only have the ability to learn new words and objects from a few examples [12–15], but they can also learn the relationship between concepts in different modalities from only a few paired examples [13]. For example, imagine a child does not know what the following concepts are: a *flower*, *bird*, *dog* or *cat*. In order for the child to learn these new concepts, he/she is shown an image of a *flower*, a *bird*, a *dog* and a *cat*. With each image the spoken word describing the object in the image is uttered. Afterwards, when asking the child to identify the image corresponding to the word "flower", the child can identify the visual instance of the *flower*. Note that this is different from the one-shot learning framework described above, in which each given example is from a single modality and paired with a label. Instead, here, a given paired example contains items from two modalities without any explicit labelling. Borovsky et al. [11] theorised that humans use specific information present in the object to learn the word,

Stellenbosch University https://scholar.sun.ac.za

and vice versa. For example, to learn the word "bird", a child might use visual information like that a bird has wings and a beak.

Using the manner in which humans learn as inspiration, Eloff et al. [1] extended unimodal few-shot learning to multimodal few-shot learning. Formally, instead of observing an item together with a class label as in unimodal one-shot learning, a multimodal one-shot learning model observes a pair of items coming from different modalities but representing the same concept. I.e. multimodal one-shot learning refers to the problem of learning new concepts from only one cross-modal paired example per concept, where each pair consists of a pair of items from different modalities but of the same concept. In the multimodal few-shot learning scenario, a model learns a new concept from a *few* cross-modal pairs of the concept. This thesis specifically considers multimodal one- and few-shot learning of spoken words and images.

1.1 Motivation

As discussed above, from only a few paired examples, children are able to learn new words using the visual objects corresponding to the word, and vice versa. They do so without having access to a lexicon representing the pronunciation of an individual word and without any transcribed speech data or labelled objects. We ask whether we can develop ML algorithms that can learn as rapidly from multimodal pairs as children.

Our multimodal few-shot learning models do not have access to transcribed speech or pronunciation dictionaries to learn acoustic phonetic units (like phonemes or even smaller units than phonemes) and sub-word structures (like syllables). Rather, these models use speech-image pairs to infer the class of a word and corresponding image. More specifically, the models learn a new class from only a few speech-image pairs of the class. As a result, these few-shot models could provide ML solutions to reduce the dependency of conventional automatic speech recognition systems on large amounts of transcribed speech data. Such large datasets of transcribed speech are expensive to collect and for many low-resource languages such datasets are not available [16]. Additionally, some low-resource languages do not have a written form [2]. In this case, it could be easier (and more affordable) to ask a native speaker of this language to give a few image examples of a word than attempting to find a written form to represent the speech data.

Practical multimodal engineering systems could also benefit from these few-shot models since they can quickly learn what the representations in different modalities of a new class look like. For example, consider being able to teach an agent like a household robot a new class by just showing the agent a visual instance while uttering the word corresponding to the visual example. Examples of systems where this new functionality could be used include Amazon's virtual artificial intelligence assistant Alexa and the Google Home system.

Stellenbosch University https://scholar.sun.ac.za

1.2 Methodology and Goals

In a multimodal speech-to-image matching task, a model should match unlabelled unseen spoken word queries to their corresponding images in a *matching set* of unseen unlabelled images. To do this, the model is only given a *multimodal support set* which contains a few speech-image pairs for each of these unseen classes. The multimodal support set can be thought of as our (small) training dataset and the matching set as our test data. The goal of this thesis is to use multimodal few-shot learning models to produce representations for spoken words and images to do this multimodal matching task using one of two approaches.

The first approach we consider is an indirect two-step approach consisting of two unimodal comparisons (a speech-speech and an image-image comparison). For these unimodal comparisons, we use separate speech and vision networks to measure similarity within a modality. To do this, the speech networks should produce similar representations for word instances of the same class and the vision networks should produce similar representations for image instances of the same class. We unpack this indirect approach in more detail in the first subsection below.

This is not an easy task since speech and image data contains a lot of information besides the class information. For the speech models to capture a spoken word's class, it should filter out nuisance information that could alter the acoustic properties of the word [17]. For example, the word "hat" and "cat" said by the same person might appear more similar than the word "hat" said by two different people. This is because spoken words contain nuisance information like speaker identity and dialect as well as channel noise which leads to acoustic variation between word instances of the same class [17, 18]. Likewise, the vision networks should filter out nuisance information when attempting to find similar representations for images of the same class. The angles and colour shades of objects in images are often nuisance information that can result in images of the same class appearing to be from different classes [4].

As our second approach, we therefore consider finding similar representations not just within a modality, but also across modalities. We do so in an attempt to remove nuisance information from the speech and image signals. This means we attempt to find similar representations in a single multimodal space for spoken words and images of the same class. By using these directly comparable speech and image representations, we can do the multimodal speech-to-image matching task using a direct approach which consists of one single direct comparison between speech queries and matching images, as outlined in more detail in Chapter 1.2.2 below.

Stellenbosch University https://scholar.sun.ac.za

1.2.1 Indirect Multimodal Few-Shot Learning

Indirect multimodal few-shot learning models consist of two unimodal networks (a speech network and a vision network) and performs the task using a multimodal support set. To do the multimodal speech-to-image matching task, the speech network is used to compare a spoken word query to each spoken word instance in the multimodal support set to find the query's closest word instance. The vision network is then used to compare the paired image of the closest word instance to each image in the matching set. The closest image is taken as the query's matching image. The speech and vision networks perform these unimodal comparisons by using a representation for each instance and calculating the cosine distance between the representations. These representations should be defined or learned in some way, and we consider a number of approaches.

The indirect approach was originally proposed by Eloff et al. [1]. We re-implement the multimodal few-shot learning study of [1], from which we build a reliable and reproducible experimental setup. Before implementing their proposed multimodal few-shot models, we implement their baseline which uses unimodal comparisons on raw speech and image data. We use dynamic time warping (DTW) on the mel-frequency cepstral coefficients (MFCCs) of spoken words for the speech-speech comparisons. For the image-image comparisons, we use cosine distance between image pixels.

For the indirect multimodal few-shot models, Eloff et al. [1] considered unimodal speech and vision classifiers and Siamese models trained in a transfer learning setting. Transfer learning is a method of training a model on a different but related dataset not containing any of the classes seen at test time [19, 20]. This training set containing the instances from different classes than the test classes, is referred to as *background data*. Unimodal classifier and Siamese models are trained on labelled background data. We then use these models to produce feature representations for the unlabelled instances of the few-shot classes (in-domain data) encountered in the matching task at test time. The hope is that by explicitly training models to find similar representations for background classes, it would also produce similar representations for unseen unlabelled data classes. The use of transfer learning can be motivated by humans using previously acquired knowledge to learn new concepts [11]. This means that when attempting to teach a child new concepts by showing the child pairs of spoken words and corresponding images of these concepts, it is conceivable that the child uses knowledge gained from previously learning other concepts to quickly learn these new concepts.

However, it is also plausible that before a child is shown new examples of visual objects paired with corresponding spoken words, the child could be exposed to a large amount of unlabelled speech and visual data from its environment. Similarly, a robotic agent could observe its surroundings, capturing unlabelled speech and image data with its sensors. Some of these unlabelled examples could correspond to the classes of the example pairs.

Stellenbosch University https://scholar.sun.ac.za

From this, we propose new indirect multimodal few-shot learning models which consists of unsupervised unimodal speech and vision networks trained on unlabelled in-domain data. In other words, we use data that contains unlabelled instances of the few-shot classes seen at test time. (However, the training instances do not occur exactly in the matching task at test time.) The hope is that the unsupervised models can infer the class information from the unlabelled data in order to find similar representations for each few-shot class.

For these unsupervised models, we use autoencoder-like network structures: the autoencoder (AE) and the correspondence autoencoder (CAE). An AE attempts to reproduce its input at its output through a bottleneck feature layer. Similarly, the CAE tries to reproduce an example of the same type or class as the input [21]. Our CAE is trained on within-modality pairs consisting of an input instance and a pair from the same class as the input instance. For the unsupervised CAEs we mine within-modality pairs in an unsupervised fashion by using cosine distance over image pixels to find image-image pairs and DTW over the MFCCs of spoken words to find speech-speech pairs.

To compare unsupervised learning to transfer learning for the indirect multimodal matching task, we also consider transfer learned variants of these unsupervised CAEs. The transfer learned variants are trained on ground truth pairs from background labelled data not containing any of the few-shot classes seen during testing. These models are new models that have not been considered before.

From our experiments, we conclude that the transfer learned models consistently outperform the unsupervised models. It is plausible that children use both previously acquired knowledge and domain specific information from unlabelled data, to learn new concepts. Therefore, we asked whether these two methodologies might be complementary: transfer learning from background data could capture general properties within a particular modality, while unsupervised learning on unlabelled in-domain data could provide a way to tailor representations to a specific test setting. We considered indirect models that consists of unimodal models that combine the unsupervised and transfer learning approaches to find representations for the unlabelled instances in the indirect matching approach. These combination models also set the groundwork for the direct multimodal few-shot learning models in the next subsection.

1.2.2 Direct Multimodal Few-Shot Learning

Our direct multimodal few-shot learning models aim to learn a multimodal embedding space from only the few speech-image pairs in the multimodal support set. This multimodal embedding space aims to map spoken words and images of the same class to similar representations in a single joint space. From these directly comparable speech and image representations, we can do the multimodal speech-to-image matching task using a direct matching approach. The direct approach consists of a single direct comparison to match

Stellenbosch University https://scholar.sun.ac.za

speech queries directly to matching images instead of using two unimodal comparisons via the support set as in the indirect approach.

We propose two new multimodal networks for the direct few-shot models: the multimodal correspondence autoencoder (MCAE) and the multimodal triplet network (MTriplet). The MCAE combines two unimodal CAEs which are connected at their bottleneck representation layers in order to find similar representations for cross-modal inputs of the same class. The MTriplet combines two triplet hinge losses. A triplet hinge loss aims to learn a relative distance metric by minimising the distance between inputs from the same class and maximising the distance between inputs from different classes. Our MTriplet aims to learn a relative distance metric between cross-modal inputs by minimising the distance between cross-modal inputs from the same class while simultaneously maximising the distance between cross-modal inputs from different classes.

Both the MCAE and MTriplet requires paired in-domain data from the speech and vision modalities. For the multimodal few-shot learning setting, we are provided with the speech-image pairs in the multimodal support set. Since this small set of speech-image pairs would not be sufficient to train a multimodal network directly, we *mine* speech-image pairs from the unlabelled in domain data. Mining is a process where we use the multimodal support set as a pivot between the unlabelled data: we consider a pair in the multimodal support set which we use to find an image from the unlabelled image dataset matching the support set pair's image instance and a word instance from the unlabelled speech dataset matching the support set pair's word instance. We then pair up this matching image and word instances to construct one mined speech-image pair. To do the comparisons between items in the support set and in the unlabelled data, we again require some way to do the comparison. For this purpose, we use transfer learned speech and vision classifiers (the best indirect multimodal few-shot model). As a result, these direct multimodal models combine unsupervised and transfer learning to learn directly comparable representations for spoken words and images from only the few example pairs in the multimodal support set. We show that this new combined direct approach outperforms all the previous models for multimodal few-shot matching.

1.3 Project Scope and Contributions

The basic goal of this study is to consider various multimodal few-shot learning models to perform a speech-to-image matching task using either an indirect two-step matching approach or a direct matching approach. To create a reliable and reproducible experimental setup to test these models, we consider the study by Eloff et al. [1] which is the first multimodal few-shot learning study and the only other study on this topic besides our work. In Chapter 3 we re-implement their proposed models and test these models on the indirect matching approach.

Stellenbosch University https://scholar.sun.ac.za

Our first contribution is a comparison of using unsupervised learning on unlabelled in-domain data vs. transfer learning on labelled background (out-of-domain) data to do indirect multimodal few-shot matching. These unsupervised indirect few-shot models which has not been considered before consists of unimodal unsupervised speech and vision AEs and CAEs trained on unlabelled in-domain data. For the comparison, we also propose new transfer learned variants of these unsupervised CAEs by training the unimodal speech and vision CAEs on labelled background data. We use these unimodal speech and vision CAEs to construct transfer learned indirect few-shot models which has never been considered before.

In Chapter 4, we evaluate these newly proposed models on the indirect matching approach and compare them to the indirect classifier and Siamese few-shot models of [1]. From this comparison, we find that the transfer learned few-shot models consistently outperform the pure unsupervised few-shot models. This leads to our second contribution in which we combine unsupervised and transfer learning to construct new indirect few-shot models. While the first combined attempt does not result in improvements, these indirect combination models set the groundwork for the direct multimodal few-shot learning models.

The direct multimodal few-shot models in Chapter 5 are our biggest contribution, since to our knowledge, this is the first direct multimodal few-shot learning study that combines the unsupervised and transfer learning methodologies. For the direct few-shot models, we use transfer learned speech and vision classifiers to automatically construct speech-image training pairs from unlabelled in-domain data in a novel pair mining scheme. The direct models outperform the indirect models on a multimodal five-shot speech-to-image matching task. We attribute this to two reasons: (1) in the direct matching approach, the word and image representations are matched using a single direct comparison instead of the two unimodal comparisons in the indirect approach which introduced a compounding of errors, and (2) by learning similar representations for cross-modal and within-modality inputs of the same class, we find representations which retains more class information and filters out more nuisance information. Overall, the MTriplet came out as our best multimodal few-shot learning model.

1.4 Thesis Overview

Chapter 2: Background. The thesis starts with a background chapter which explains neural network fundamentals and the datasets we use throughout the thesis. This chapter also discusses background literature and theory which focusses on unimodal few-shot learning. Note that each of the subsequent chapters contains an introductory section covering literature relevant to the content covered in that chapter.

Chapter 3: Multimodal Few-Shot Learning using Transfer Learning. We

Stellenbosch University https://scholar.sun.ac.za

extend unimodal few-shot learning to multimodal few-shot learning and devote the entire chapter to fully investigate, re-implement and improve the multimodal few-shot learning study done by Eloff et al. [1]. This study, which proposed using transfer learned indirect multimodal few-shot models, forms the basis of our work.

Chapter 4: Unsupervised vs. Transfer Learning for Multimodal Few-Shot Learning. After implementing the indirect few-shot models proposed by Eloff et al. [1], we propose unsupervised indirect few-shot models and additional transfer learned variants of these unsupervised models. We conclude this chapter by considering indirect models which are combinations of unsupervised and transfer learning. This lays the groundwork for our direct multimodal few-shot learning models discussed in Chapter 5.

Chapter 5: Direct Multimodal Few-Shot Learning. We combine unsupervised and transfer learning to obtain direct multimodal few-shot learning models. The direct models aim to find direct mappings between spoken words and corresponding images so that we can use these models in a direct multimodal speech-to-image matching approach.

Chapter 6: Summary and Conclusions. This thesis is concluded with a summary of what we did and all the conclusions we made throughout the thesis. In this chapter, we also discuss possible future work which could build on the work in this thesis. We specifically give recommendations in order to extend this work to be applicable in practical settings.

1.5 Code and Publications

The comparison of unsupervised and transfer learning models of Chapter 4 was submitted to and accepted at *Interspeech 2020* in a paper entitled *Unsupervised vs. transfer learning for multimodal one-shot matching of speech and images*. We release the corresponding source code for the experiments in Chapter 3 and 4 at :

`https://github.com/LeanneNortje/multimodal_speech-image_matching`.

We release source code for the experiments of Chapter 5 at :

`https://github.com/LeanneNortje/direct_multimodal_few-shot_learning`.

Stellenbosch University https://scholar.sun.ac.za

Chapter 2

Background

This background chapter will cover concepts that the reader should be familiar with in order to follow the subsequent chapters. Chapter 2.1 introduces unimodal few-shot learning which is the key idea from which multimodal few-shot learning is developed in Chapter 3. In Chapter 2.2 we describe the speech and image datasets used throughout this thesis. Chapter 2.3 describes different types of networks we use to build our models and the fundamental algorithms used to train these models. Particulars regarding the implementation of our models, as well as the resources to reproduce the results of Chapters 3 to 5, are given in Chapter 2.4.

2.1 Unimodal Few-Shot Learning

Unimodal one-shot learning refers to the problem of learning new concepts from only *one* labelled example per concept in a single modality. For the unimodal *few-shot* learning setting, a *few* labelled examples per concept are given instead of just one.

> **Unimodal Few-Shot Learning** entails learning new concepts from only a few labelled examples per concept in a single modality.

We use blocks such as the one above to define specific concepts throughout this thesis. In Chapter 2.1.1 we first explain the unimodal few-shot learning task, and then give one approach to do this task. Chapter 2.1.2 gives more background on the origin of unimodal few-shot learning and previous approaches for performing this task.

2.1.1 Unimodal Speech or Image Few-Shot Classification

Unimodal few-shot classification is a task in which a model is prompted to match an unlabelled unseen query to its corresponding label after only seeing a *unimodal support set* $\mathcal{S}$. A unimodal support set contains data examples from a single modality where each example is tagged with a text label. Unimodal few-shot classification can be considered

Stellenbosch University https://scholar.sun.ac.za

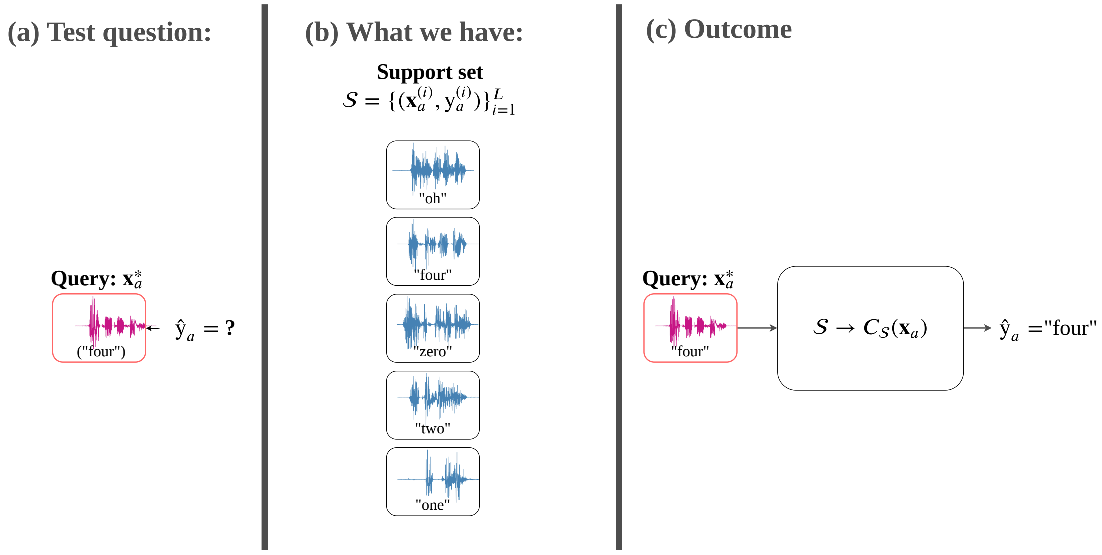


Figure 2.1: Unimodal one-shot speech classification leads to (a) the question shown at test time. To answer this question, a model is only given (b) a supports set to (c) predict the query's class.

for any modality. For illustrative purposes, we start with the example of few-shot speech classification and then also explain few-shot image classification.

During a few-shot speech classification task, a speech model is presented with an unseen unlabelled speech query $\mathbf{x}_a^*$ and prompted to match the query to its corresponding label as illustrated in Figure 2.1(a). For the *one-shot* setting, the model is shown a unimodal one-shot support set $\mathcal{S}$ containing *one* isolated spoken word $\mathbf{x}_a^{(i)}$ with a text label $\mathrm{y}_a^{(i)}$ for each of the L word classes as shown in Figure 2.1(b). Although Figure 2.1(b) only illustrates $L = 5$, the speech digit classes has eleven possible classes including "one" to "nine" as well as "oh" and "zero" since the words "oh" and "zero" both refers to the digit *0*. None of the speech queries $\mathbf{x}_a^*$ occurs exactly in the support set. From this support set, the model should learn a classification metric $C_{\mathcal{S}}$ that can make predictions on an unlabelled unseen test query $\mathbf{x}_a^*$ as shown in Figure 2.1(c).

One approach to do the speech classification task is to simply use unimodal comparisons to compare a query $\mathbf{x}_a^*$ to each item in the support set and then predict the query's label as the label of the closest item, as illustrated in Figure 2.2(a). In our speech classification example, the classifier $C_{\mathcal{S}}$ used to do this classification task is an L-way one-shot speech learning model which consists of an L-way one-shot speech support set and a speech model capable of measuring within-modality similarity. This is just one (common) approach to solve this task. Various others approaches are mentioned in the next section.

We can extend one-shot learning to K-shot learning. For unimodal L-way K-shot learning, the unimodal support set $\mathcal{S}$ contains L classes and K labelled examples per class. Throughout the thesis we use K-shot and few-shot interchangeably.

The one-shot image classification task shown in Figure 2.3 is done similarly as the one-shot speech classification task in Figure 2.2. However, instead of using speech examples

Stellenbosch University https://scholar.sun.ac.za

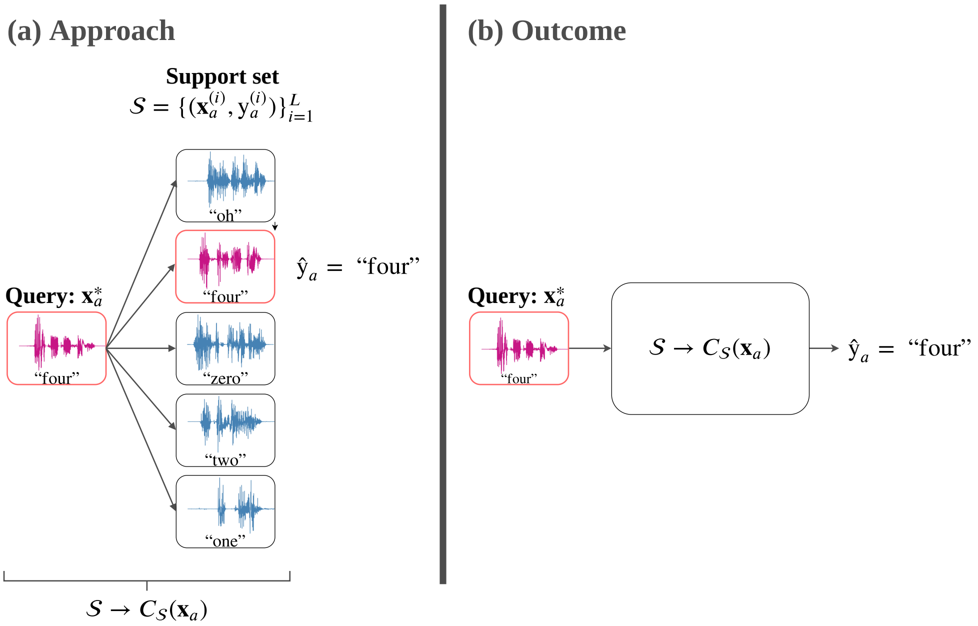

(a) Approach
Support set
$\mathcal{S} = \{(\mathbf{x}_a^{(i)}, y_a^{(i)})\}_{i=1}^{L}$
"oh"
"four"
"zero"
"two"
"one"
Query: $\mathbf{x}_a^*$
"four"
$\hat{y}_a$ = "four"
$\mathcal{S} \rightarrow C_{\mathcal{S}}(\mathbf{x}_a)$
(b) Outcome
Query: $\mathbf{x}_a^*$
"four"
$\mathcal{S} \rightarrow C_{\mathcal{S}}(\mathbf{x}_a)$
$\hat{y}_a$ = "four"

Figure 2.2: One approach to do the speech classification task, is to find (b) a metric $C_{\mathcal{S}}$ to classify given speech inputs by (a) using the label of a query's neighbour in the support set.

we use image examples: a vision model is shown an unlabelled unseen image query $\mathbf{x}_v^*$ and is prompted to match the image query to its corresponding label. To do this, the model is only given a unimodal K-shot image support set $\mathcal{S}$ which contains K image instances $\mathbf{x}_v^{(i)}$ labelled with a text class label $y_v^{(i)}$ for each of the L classes. Although Figure 2.3 illustrates a support set with only five classes, there are ten possible image digit classes (*0* to *9*). We specifically do this task by comparing $\mathbf{x}_v^*$ to each image instance $\mathbf{x}_v^{(i)}$ in $\mathcal{S}$ and the label of its closest item is taken as the image query's predicted label.

A unimodal K-shot L-way classification task is implemented with *unimodal episodes*

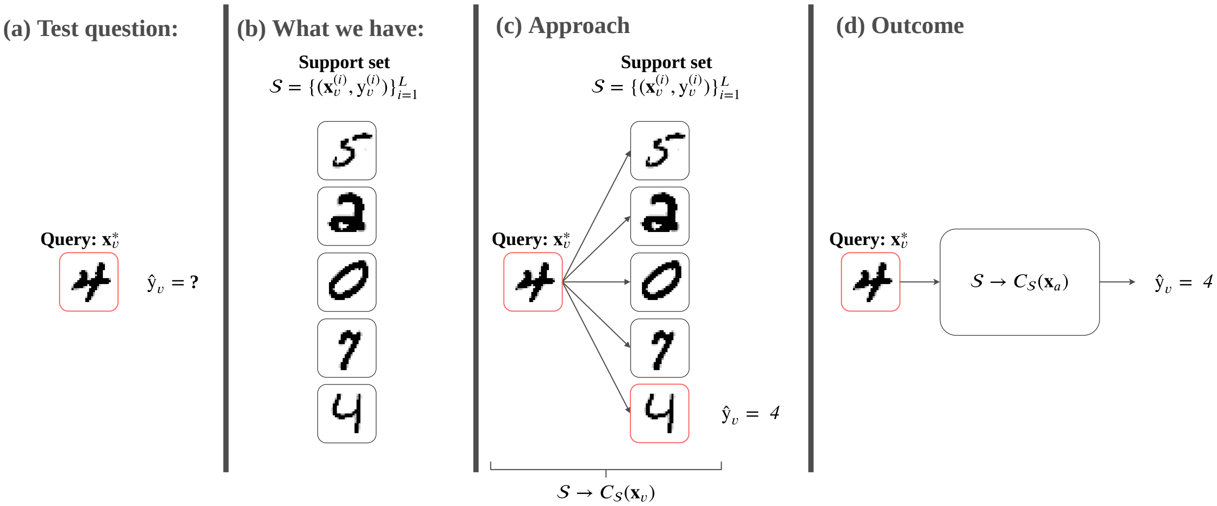

(a) Test question:
Query: $\mathbf{x}_v^*$
$\hat{y}_v$ = ?
(b) What we have:
Support set
$\mathcal{S} = \{(\mathbf{x}_v^{(i)}, y_v^{(i)})\}_{i=1}^{L}$
(c) Approach
Support set
$\mathcal{S} = \{(\mathbf{x}_v^{(i)}, y_v^{(i)})\}_{i=1}^{L}$
Query: $\mathbf{x}_v^*$
$\hat{y}_v$ = 4
$\mathcal{S} \rightarrow C_{\mathcal{S}}(\mathbf{x}_v)$
(d) Outcome
Query: $\mathbf{x}_v^*$
$\mathcal{S} \rightarrow C_{\mathcal{S}}(\mathbf{x}_a)$
$\hat{y}_v$ = 4

Figure 2.3: Unimodal one-shot image classification where (c) illustrates how the model makes its prediction for (a) the question shown at test time by using the support set in (b). The outcome (d) is a metric $C_{\mathcal{S}}$ to classify given image inputs.

Stellenbosch University https://scholar.sun.ac.za

so that each episode is an instance of a K-shot classification task. Vinyals et al. [8] was the first to use unimodal K-shot episodes and defined each episode to contain a set of queries – a so called *query set* – and a unimodal K-shot L-way support set. The instances in the query set and support set are from the same modality, e.g. either the vision or speech modality. For each episode, the K-shot model under consideration is prompted to match each query in the episode's query set to its corresponding label in the episode's K-shot support set. Various studies uses this episode-idea to train few-shot learning models [8, 10, 22–25]. However, in our few-shot experiments we do not explicitly train on any episodes.

In Chapter 3 we explain how unimodal few-shot learning can be extended to *multimodal few-shot learning*, specifically for speech and images in our case. To implement multimodal few-shot learning we extend the unimodal episodes to *multimodal episodes* as discussed in Chapter 3.

2.1.2 Related Work

Our focus is not an exhaustive comparison between different models for unimodal few-shot learning, but rather the extension of unimodal to multimodal few-shot learning. The goal of this section is therefore to discuss unimodal few-shot studies that followed approaches relevant to the approaches we use.

This section will mainly discuss speech or vision one- or few-shot learning since these are the two modalities of interest to us. However, it is important to note that unimodal one- or few-shot learning can be done in any modality like gesture recognition [26–28], video [29] and robotics [30, 31].

2.1.2.1 Transfer Learning with Classifiers

Although a few researchers investigated (unimodal) one-shot learning as early as the 1980's and 1990's, the groundbreaking ML methods to do one-shot learning was developed in the early 2000's [7]. In [3] and [4], Fei-Fei et al. uses *transfer learning* to construct a variational Bayesian framework for one-shot learning of objects in images.

Transfer learning is a method that leverages existing data by training a model on a different but related set of classes than the classes seen during test time [19, 20].

This training dataset containing items from different classes than the classes seen at test time, is referred to as *background data*.

In Chapter 3 and 4, we consider separate speech and vision transfer learned models trained on background data containing a large amount of classes. However, Fei-Fei et al. [3, 4] only considers transfer learned vision models trained on images of objects from only three background classes. For each of these three classes they train a probabilistic classifier

Stellenbosch University https://scholar.sun.ac.za

to predict the probability that an input image is of this class. To get a probabilistic model capable of classifying a new unseen class, they average the model parameters of these three models. Differently to our approach, they then update the model on the one or few labelled image examples given for this new unseen class. When encountering an unlabelled unseen image query, this new model predicts the probability that the query belongs to the few-shot class. The probabilistic model in [3] is extended in [4] to enable the model to find a classifier for a new unseen class from a larger number of background object classifiers.

A collection of studies [32, 33] by Lake et al. investigated compositional generative Bayesian models to do one-shot learning of the Omniglot character images (which we also use and introduce in more detail in Chapter 2.2.2.2). To do one-shot learning, their Bayesian models use each character's motor data [5, 34], where the motor data of a character is the strokes used to write a character, i.e. the order, composition and direction in which the subparts of a character is written. The probabilistic models in [32] and [33] decompose the characters into common primitive subparts and generates new characters from these primitive subparts.

Lake et al. [33] uses Bayesian program learning to learn a character classifier based on a character's unique composition and the relation between its primitive subparts. Similarly, Lake et al. [32] uses hierarchical Bayesian program learning (HBPL) to learn a character classifier by using the composition of a character and the order in which its subparts are drawn. Similarly as our transfer learned vision models, both of these transfer learned Bayesian classifiers are trained on background character classes from the Omniglot dataset. However, since we do not have motor data for our few-shot digit classes, we only train our models on the character images so that we can apply our models to the unseen few-shot classes at test time. During the classification task, each Bayesian model of Lake et al. [32, 33] calculates a classification score between the strokes present in the unlabelled unseen query and each of the one-shot labelled character examples by using the motor data of these instances. The query is classified according to the class of the one-shot character image with the highest score to the query.

From the studies discussed in this section and in the next three subsections, it is clear that transfer learning is a common approach to do one- or few-shot learning. In Chapters 3 to 5 we will use transfer learning either in its pure form or in combination with unsupervised learning to find multimodal few-shot learning models. Transfer learning can be motivated by the observation that humans can call on prior knowledge when learning new concepts [11].

2.1.2.2 Metric Learning

By utilising transfer learning, Koch [7] trains Siamese neural networks to distinguish between features of background same/different pairs of handwritten characters in the Omniglot dataset. Similarly to the approach we use in Chapter 3, [7] uses these Siamese

Stellenbosch University https://scholar.sun.ac.za

networks in the hope that these embedding models would generalise to such an extent that it will also generalise to unseen classes for one-shot learning. Koch [7] specifically attempts to find a Siamese network that maps the input images of an unseen one-shot class to similar embeddings, i.e. a similarity metric that applies to unseen classes.

> **Metric Learning** attempts to map inputs to an embedding space where the embeddings of similar inputs are close to each other and the embeddings of different inputs are far apart [35].

To learn this metric, [7] uses a modified L_1 distance loss function where we instead use a triplet hinge loss.

Is such a metric learning approach more beneficial to do few-shot learning than the Bayesian approaches used in the previous subsection? For a one-shot task on the same Omniglot test subset, [7] found that the Siamese network with one-shot accuracy of 92.0% were outperformed by a HBPL model with a one-shot accuracy of 95.2%. From this it seems like metric learning is inferior to a probabilistic classification approach. However, the matching network proposed by Vinyals et al. [8] which learns a metric to do few-shot learning, outperformed both the Siamese and probabilistic approach by achieving a one-shot accuracy of 98.1% on the same Omniglot test subset.

Specifically, Vinyals et al. [8] used transfer learned matching networks to learn a metric to do few-shot learning of either objects in ImageNet images or Omniglot character images. The matching networks attempt to learn how to learn in the rapid few-shot manner required to do few-shot learning at test time. This is done by using advances in attention and memory to enable this rapid learning, as well as explicitly training the models using background image classes on the same few-shot image classification task described in Chapter 2.1.1. Similarly to our approach, a matching network can be seen as a weighted nearest neighbour classifier since it learns embedding functions to find similar embeddings for a query and its few same class instances in the support set. However, we do not explicitly train our models on background few-shot tasks. They evaluate the matching network trained on background Omniglot classes by using few-shot image classification tasks on two datasets: the Omniglot test subset and the completely disjoint MNIST dataset which contains our few-shot digit classes (we also use the MNIST dataset and introduce it in more detail in Chapter 2.2.2.1).

Using the same few-shot task on the Omniglot test subset as Vinyals et al. [8], Snell et al. [10] proposed prototypical networks for few-shot learning which achieved a one-shot accuracy of 98.8%. Therefore, this approach followed by [10] improves the matching network approach of [8] which achieved an accuracy of 98.1%. From these two studies, we conclude that metric learning is an appropriate avenue to pursue for few-shot learning.

Similarly to us, Salakhutdinov et al. [36] uses the MNIST dataset as their few-shot digit classes. More specifically, Salakhutdinov et al. [36] performs one-shot learning of

Stellenbosch University https://scholar.sun.ac.za

objects in natural images or handwritten MNIST digit images. To do one-shot learning, [36] uses a transfer learned hierarchical nonparametric Bayesian model trained on classes not seen at test time. For example, they would train a model on the digit classes *0* to *8* and use the class *9* as the few-shot class. In contrast, we use all the digit classes as our few-shot classes. We either train transfer learned vision models on background data not containing any of digit classes or unsupervised unimodal or multimodal models trained on unlabelled instances of the digit classes. For both of these approaches, we do not update our models on the given examples of the few-shot classes. However, during the one-shot classification task, Salakhutdinov et al.'s [36] Bayesian model uses the labelled example of an unseen one-shot class to fine-tune a similarity metric for this new class.

2.1.2.3 Meta-Learning

On the same few-shot task approach as Vinyals et al. [8], Ravi and Larochelle [23] used meta-learning to find a model capable of few-shot image learning.

> **Meta-Learning** is to train a model referred to as a *meta-learner* on a variety of learning tasks. The meta-learner is then used to train another model referred to as the *learner* which is capable of learning a variety of new tasks from a small number of training examples [25].

Ravi and Larochelle [23] trains a long short-term memory (LSTM) meta-learner neural network which learns the exact optimisation required to train a learner that can be used for few-shot learning. They use a classifier for the learner network.

Santoro et al. [24] used a memory-augmented meta-learner neural network trained on background character classes to do few-shot character classification on the the Omniglot test subset at test time. Mishra et al. [37] follows the same apporach as Santoro et al. [24], but instead of a memory-augmented meta-learner, Mishra et al. [37] uses a meta-learner network consisting of temporal convolutions and attention optimisation to find a learner capable of few-shot character learning.

Finn et al. [31] extended the meta-learning approach to be model-agnostic: model-agnostic meta-learning (MAML) uses the meta-learner to learn the standard model parameters for any gradient decent trained model or learning problem. These standard model parameters are then used for the learner network from which fast adaption (by updating the model with the few-shot test classes) is possible. For the few-shot character classification task on Omniglot images, MAML achieved a one-shot accuracy of 98.7% and therefore outperforms the matching network approach with a one-shot accuracy of 98.1% [8]. However, this MAML approach just falls short of the prototypical network approach with a one-shot accuracy 98.8% [10].

Although meta-learning has received significant attention for few-shot learning, recent work by Tian et al. [38] shows that a simpler metric learning approach for few-shot

Stellenbosch University https://scholar.sun.ac.za

image learning outperforms these complicated meta-learning approaches like MAML. On background data, they trained a simple embedding model on the same few-shot task as Vinyals et al. [8]. On top of the embeddings learned by the embedding model, they train a classifier. The aim is to get good embeddings so that the embedding model produces similar embeddings for unseen classes to do the few-shot classification task in a similar setup as Vinyals et al. [8] (Chapter 2.1.1). We conclude that complex meta-learning approaches are not necessary to do few-shot learning. Therefore, in Chapter 3 we will focus on classification networks and compare these networks to Siamese networks which is a well-established metric learning approach. Nevertheless, in Chapter 6 we note that a thorough comparison could be useful in future work.

2.1.2.4 Unimodal Speech Classification

Although few-shot image classification has received significant attention, studies into few-shot speech classification is limited. We will now consider two few-shot speech studies: one study that uses transfer learning in a probabilistic approach and one that uses transfer learning together with a metric learning approach.

Lake et al. [6] considers one-shot learning in the speech domain with a hierarchichal hidden Markov model (HHMM). The HHMM utilises the composition of phoneme-like (acoustic) units that each word in an utterance consists of. Two HHMMs are trained using transfer learning whereafter both models are tested on one-shot learning of Japanese spoken words. One HHMM is trained on background English speech data and the other one on background Japanese speech data that does not contain any of the Japanese one-shot word classes seen at test time.

We also train transfer learned speech networks on background data not containing any of the few-shot digit classes seen at test time. We use these models to generalise to new unseen classes by attempting to find similar embeddings for all the speech instances of the same few-shot class. An unseen query can then be classified according to the label of its closet embedding in the support set. In contrast, for the one-shot classification task using each HHMM, the HHMM calculates a classification score between the unlabelled unseen word query and each of the one-shot labelled word examples. The classification score is calculated based on the ten most likely acoustic units in each word. The query is classified according to the class label of the one-shot word example with the highest classification score to the query.

On this task, the Japanese trained HHMM came closer to human performance than the English trained HHMM. Lake et al. [6] concluded that the transfer of knowledge between two data domains that contains more common properties (e.g. language) works better for one-shot learning than two completely different data domains (e.g. from different languages). Therefore, for our few-shot speech models, we ensure the training data and few-shot test classes are from the same language (English).

Stellenbosch University https://scholar.sun.ac.za

The same few-shot speech classification task approach we use for our speech models, is also used by Parnami and Lee [22]. This few-shot speech task is simply the speech version of the few-shot image task defined by Vinyals et al. [8]. Parnami and Lee [22] specifically uses prototypical networks to learn a metric to do few-shot speech learning. Similarly to the prototypical networks of Snell et al. [10] discussed above, they use a prototype representation for each few-shot class which is the mean across the few labelled examples given for the few-shot class. They use prototype representations for each few-shot class in the support set instead of using the few examples separately like we do. Differently to our speech models, [22] trains these prototype networks using few-shot speech classification tasks on background data. At test time, these models are then used to few-shot learn spoken keywords to do keyword spotting.

2.2 Data

This section discusses the speech (Chapter 2.2.1) and image (Chapter 2.2.2) datasets used throughout the thesis, as well as the data processing done before training. Chapter 2.2.3 states some general details regarding the manner in which we use the data.

2.2.1 Speech Data

We parametrise all speech data, before training, with our version of Python speech features [39] as mel-frequency cepstral coefficients (MFCCs) [40] using a window length of 25 milliseconds and a frame shift of 10 milliseconds. For each speech dataset, we perform per speaker normalisation of the speech segments. By using existing forced alignments, we split the speech sequences into isolated words. All speech models are trained on isolated spoken words where each word is represented using static MFCCs. However, first and second order derivatives are used in the DTW baseline of Chapter 3 where it is beneficial.

2.2.1.1 In-Domain Speech Data: TIDigits

The TIDigits corpus which contains eleven digit classes: “one” to “nine” as well as “oh” and “zero” since the words “oh” and “zero” both refers to the digit *0*. We use the TIDigits corpus as our in-domain speech data. This means that we use the digit classes in this dataset as the few-shot classes in all subsequent chapters.

The corpus consists of spoken digit sequences from 326 speakers with each speaker uttering 77 digit sequences [41]. The spoken digit words in each sequence are isolated. A few of these isolated spoken digit words are shown in Figure 2.4. We further divide these isolated words into training, validation and test subsets with no speaker overlap between subsets which results in an average of 1931 training instances per word class.

Stellenbosch University https://scholar.sun.ac.za

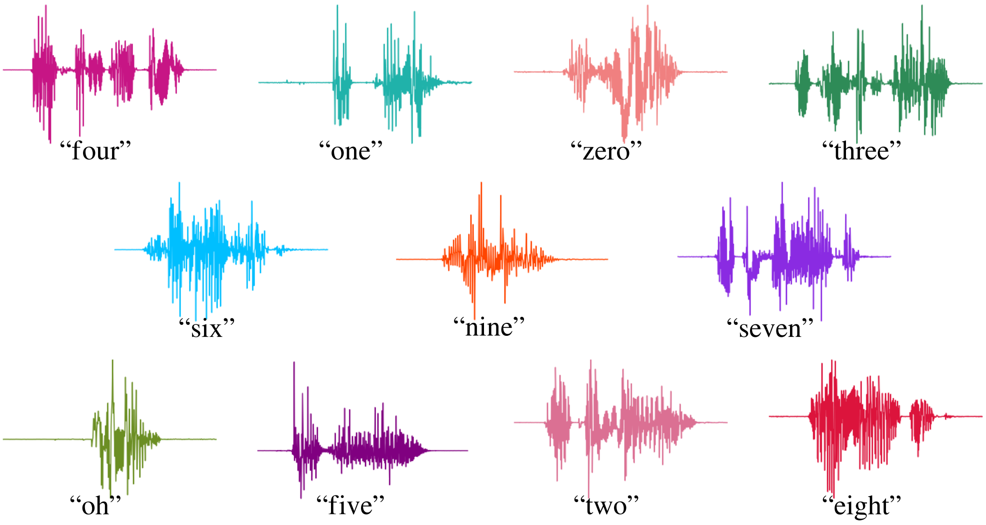


Figure 2.4: A few speech samples from the TIDigits corpus. For illustrative purposes we show labels for the spoken word instances: in most of our experiments (apart from the diagnostic experiments), no labels are used since the digit labels are our few-shot classes.

2.2.1.2 Background and Developmental Speech Data: Buckeye

For background and developmental speech data, we use the Buckeye corpus of English conversational speech from 40 speakers [42]. As explained in Chapter 2.1.2.1, background data does not contain any of the few-shot digit classes. We use this background speech data to train transfer learned speech models. Development procedures are outlined in Chapters 3 to 5, but in short we use the development data to fine-tune model hyperparameters and hard restrictions used to mine pairs.

Figure 2.5 shows some spoken word examples that occur in the Buckeye corpus, as well as the process we follow to isolate spoken words using forced alignments of the utterances contained in the TIDigits and Buckeye datasets. After isolating all the spoken words in the Buckeye corpus, the corpus contains 8280 different word classes.

We divide the isolated words into training, validation and test subsets with no class overlap between subsets and ensure that there are no instances of the above target digit

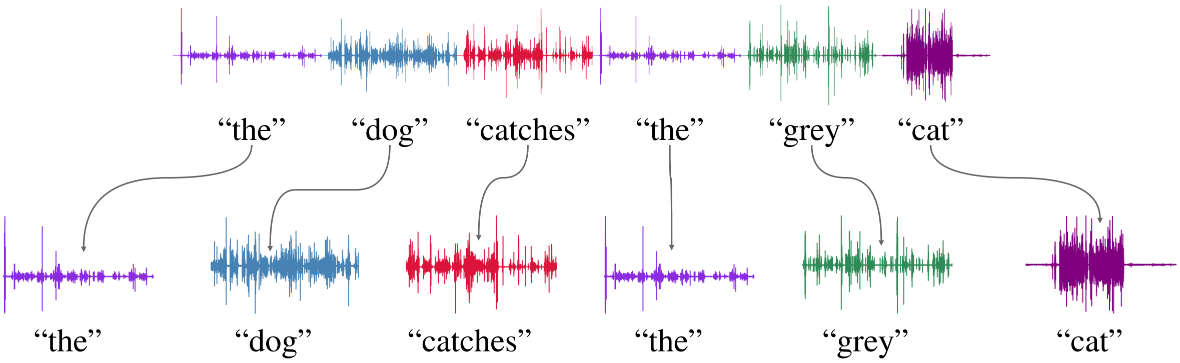


Figure 2.5: Isolating the words in the conversational spoken sequences of the Buckeye corpus.

Stellenbosch University https://scholar.sun.ac.za

classes in the background speech data. This results in an average of 25 training instances per word class.

It is important to note that the Buckeye corpus (our background data) contains a large number of classes with only about 25 examples per class, whereas the TIDigits corpus (with our few-shot classes) contains only eleven word digit classes with a large number of examples per class.

2.2.2 Image Data

We normalise all pixels of images in the MNIST image dataset discussed in Chapter 2.2.2.1 and the Omniglot image dataset discussed in Chapter 2.2.2.2 to be in the range of $[0, 1]$ and we ensure that all images are 28×28 pixels.

2.2.2.1 In-Domain Image Data: MNIST

As our in-domain image data, we use the MNIST corpus which contains 28×28 grayscale handwritten digit images from the ten digit classes (*0* to *9*) [43]. This means we use the digit classes in this dataset as the vision instances of our few-shot classes in the subsequent chapters. A few of the digit images in the corpus are shown in Figure 2.6. The MNIST images in Figure 2.6 are inverted variants of the actual MNIST images. We divide the corpus into training, validation and testing subsets which results in an average of 5500 training instances per digit class.

2.2.2.2 Background and Developmental Image Data: Omniglot

For background and developmental image data we use the Omniglot corpus of handwritten characters [33]. We use this background image data (that does not contain any image

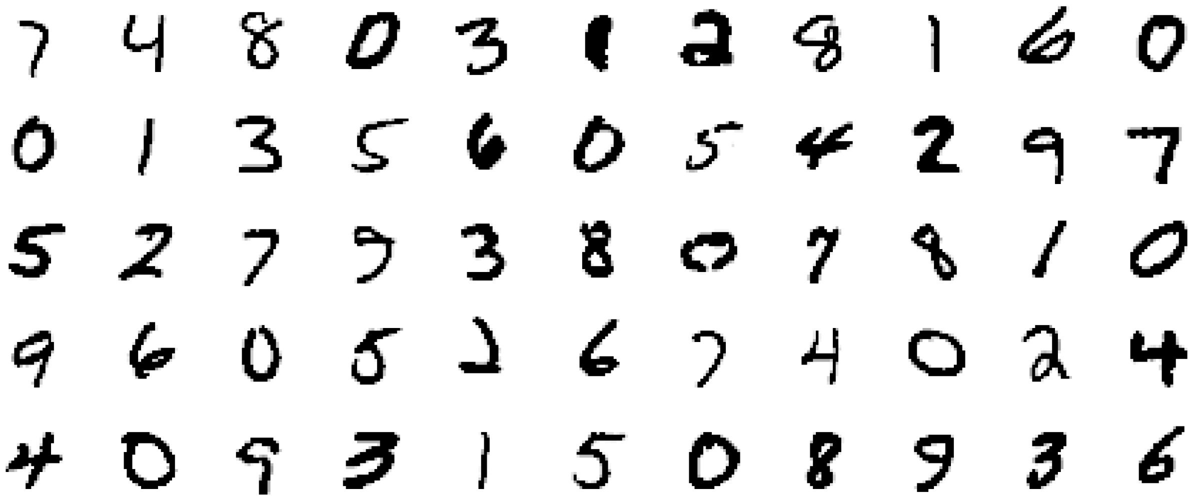


Figure 2.6: A few digit image examples from the MNIST corpus.

Stellenbosch University https://scholar.sun.ac.za

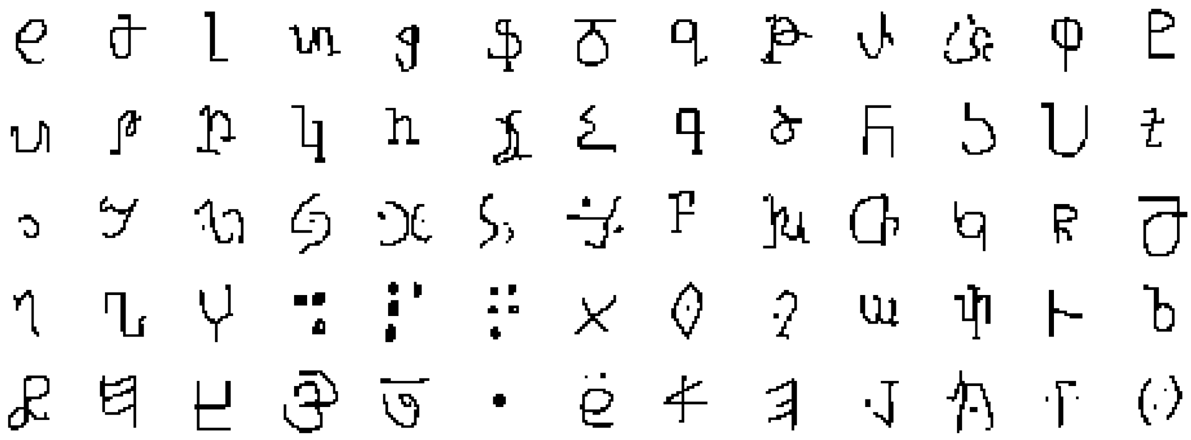


Figure 2.7: A few character images that occur in the Omniglot corpus.

instances of digit classes) to fine-tune the hyperparameters of the vision models and to fine-tune the hard restrictions we use to mine pairs.

The corpus contains characters from 50 different alphabets ranging from common languages like Latin to less common local dialects, as well as fictitious character sets like Klingon [7]. The number of character classes per alphabet differs between 15 and 40 with only 20 examples per character class. This results in 1623 different character classes which we invert and downsample to 28 × 28 pixels. A few Omniglot image examples are shown in Figure 2.7.

In order to use the Omniglot corpus for background modelling, it needs to be in the same format as the MNIST data. This would, for instance, allow a transfer learning model trained on Omniglot to be used to process images from MNIST. To make the corpora compatible, we therefore downsample the Omniglot images to have the same dimension as the MNIST images. Furthermore, we invert the Omniglot images to use the same convention as the MNIST images where a 1 indicates a white pixel and a 0 indicates a black pixel. This ensures that the vision models trained on Omniglot is compatible to the MNIST images seen at test time. It is important to note that the Omniglot corpus contains 1623 different character classes with only 20 examples per class, whereas the MNIST corpus contains only ten digit classes with a large number of examples per class.

We divide Omniglot into training, validation and testing subsets with no class overlap between subsets and we ensure that there are no instances of the target digit classes in the background image data.

2.2.3 Data Usage

It is important to note that although we use labelled in-domain datasets (TIDigits and MNIST), we use them in an unlabelled manner for unsupervised or few-shot learning setups. All few-shot matching experiments are performed on the MNIST and TIDigits test subsets. This enables us to get accuracy scores for the unimodal and multimodal few-shot

Stellenbosch University https://scholar.sun.ac.za

tasks (discussed in Chapter 3 and Chapter 5) by comparing predicted labels to the actual labels.

2.3 Neural Networks

In this section we introduce the reader to the fundamentals of neural networks. This is not an exhaustive discussion of the topic, we simply cover the fundamentals so that the reader can follow subsequent chapters. These neural networks are used throughout the thesis.

Here we use the notation where a neural network is represented by a function $g_{\mathbf{\Phi}}(\mathbf{x})$, where $\mathbf{\Phi}$ is the trainable parameters of the network and $\mathbf{x}$ some input to the network. Feedforward neural networks (FFNNs) and convolutional neural networks (CNNs) (Chapter 2.3.1 and 2.3.2 respectively) are appropriate choices to model fixed-length inputs to fixed-size feature embeddings. Recurrent neural networks (RNNs) (discussed in Chapter 2.3.3) are more appropriate to model variable-length inputs to fixed-size feature embeddings.

A single $g_{\mathbf{\Phi}}(\mathbf{x})$ function representing either a FFNN or a CNN or an RNN can be used as part of a much larger model $f_{\mathbf{\Theta}}(\mathbf{x})$, where $\mathbf{\Theta}$ is the trainable parameters of the entire model. Multiple $g_{\mathbf{\Phi}}(\mathbf{x})$ functions can be used in various different configurations to construct a model function. For example, $f_{\mathbf{\Theta}}(\mathbf{x})$ can be a chain of different $g_{\mathbf{\Phi}}(\mathbf{x})$ functions representing different network types:

$$f_{\mathbf{\Theta}}(\mathbf{x}) = g_{\mathbf{\Phi}_{\text{RNN}}}(\, g_{\mathbf{\Phi}_{\text{FFNN}}}(\, g_{\mathbf{\Phi}_{\text{CNN}}}(\,\mathbf{x}\,)\,)\,), \tag{2.1}$$

where $\mathbf{x}$ is the input to the model.

The shape and objective functions of a classifier, a Siamese neural network, an AE and a CAE are considered in Chapters 2.3.4 to 2.3.7. An objective function sets out some specific constraints for the model targets (outputs) which forces the model to learn some internal structure that can produce targets meeting these constraints. The objective function is also called a loss function which we denote as ℓ. Although other studies may add different meanings to these two terms, we use them interchangeably. Finally, Chapter 2.3.8 explains the algorithm used to train a model by using its specified loss function ℓ to learn a suitable $f_{\mathbf{\Theta}}(\mathbf{x})$.

2.3.1 Feedforward Neural Networks

The information in a feedforward neural network (FFNN) flows from the network's input $\mathbf{x}$ to its output layer $\hat{\mathbf{y}}$, without any feedback [44]. The network consists of a certain amount of intermediate layers referred to as *hidden layers*, between the network's input $\mathbf{x}$ and its output $\hat{\mathbf{y}}$. Each hidden layer consists of a weight matrix $\mathbf{W} \in \mathbb{R}^{(N \times M)}$ and a bias vector

Stellenbosch University https://scholar.sun.ac.za

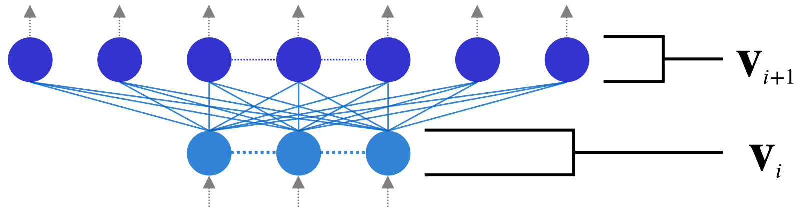


Figure 2.8: A FFNN consists of a number of fully connected layers.

$\mathbf{b} \in \mathbb{R}^{(N \times 1)}$. The layer output is a vector $\mathbf{v} \in \mathbb{R}^{(N \times 1)}$ calculated as

$$\mathbf{v} = \sigma(\mathbf{W}\mathbf{u} + \mathbf{b}), \tag{2.2}$$

where σ is some activation function and $\mathbf{u} \in \mathbb{R}^{(M \times 1)}$ is the input vector to the layer. Common activation functions include the sigmoid, tanh and ReLU functions. We mostly use the ReLU activation function.

The hidden layers are called *fully connected layers* since each unit in the layer output $\mathbf{v}$ is connected to each unit in the layer input $\mathbf{u}$. Throughout this thesis, the chosen output layer dimension N of a fully connected layer is given in the following format in figures depicting model architectures: (N).

Multiple hidden layers can be connected by giving the output of one layer as the input to another layer as illustrated in Figure 2.8. By connecting any number of hidden layers, we construct a FFNN function $g_{\mathbf{\Phi}_{\text{FFNN}}}(\mathbf{x})$. The input $\mathbf{x}$ to the FFNN is given as the input to the first hidden layer and the last layer's output is taken as the FFNN's output $\hat{\mathbf{y}}$. We group all the trainable parameters of a FFNN under the variable $\mathbf{\Phi}$. The trainable parameters of the entire network includes the trainable parameters $\mathbf{W}$ and $\mathbf{b}$ of each hidden layer in the network.

2.3.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are neural networks used for processing matrix-like data [44] and consists of one or multiple *convolutional layers*. A convolutional layer consists of N trainable filters $\mathbf{f} \in \mathbb{R}^{(\text{height} \times \text{width})}$ (also called kernels). Figure 2.9 shows the N filters are a set of weights so that one of these filters $\mathbf{f}$ connects each unit in a so-called feature map $\mathbf{v} \in \mathbb{R}^{(N_1 \times N_2)}$ to local patches in the input $\mathbf{u} \in \mathbb{R}^{(M_1 \times M_2)}$ to the layer [44]. Each unit in the feature map is calculated with convolution using a particular (height × width) filter:

$$\mathbf{v}[a, b] = \sum_{i}^{\text{height}} \sum_{j}^{\text{width}} \mathbf{u}[i, j]\ \mathbf{f}[a - i, b - j], \tag{2.3}$$

where a and b are some index positions corresponding to a unit in the feature map. The output of a convolutional layer is therefore N feature maps $\mathbf{v}$ where each of the N filters $\mathbf{f}$ produces a feature map from the input $\mathbf{u}$. The filter dimensions and number of filters for

Stellenbosch University https://scholar.sun.ac.za

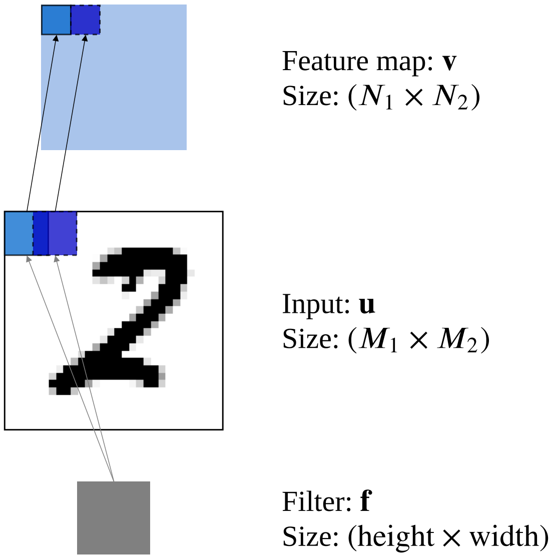


Figure 2.9: The convolutional calculation of a convolutional layer.

a convolutional layer will be given in the following format in the figures throughout the thesis: (height × width × N).

Convolutional layers are connected by giving the output feature maps of one layer as the input to a subsequent layer. By connecting various convolutional layers we can construct a CNN function $g_{\mathbf{\Phi}_{\text{CNN}}}(\mathbf{x})$ from the network input $\mathbf{x}$ to its output $\hat{\mathbf{y}}$, where $\mathbf{\Phi}$ is a grouping of the entire CNN's trainable parameters. These trainable parameters include each convolutional layer's N filters $\mathbf{f}$. The network's input $\mathbf{x}$ is the input to the first convolutional layer and the network's output $\hat{\mathbf{y}}$ is the feature maps produced by the last convolutional layer.

2.3.3 Recurrent Neural Networks

Recurrent neural networks (RNNs) are used to process sequential data by using some

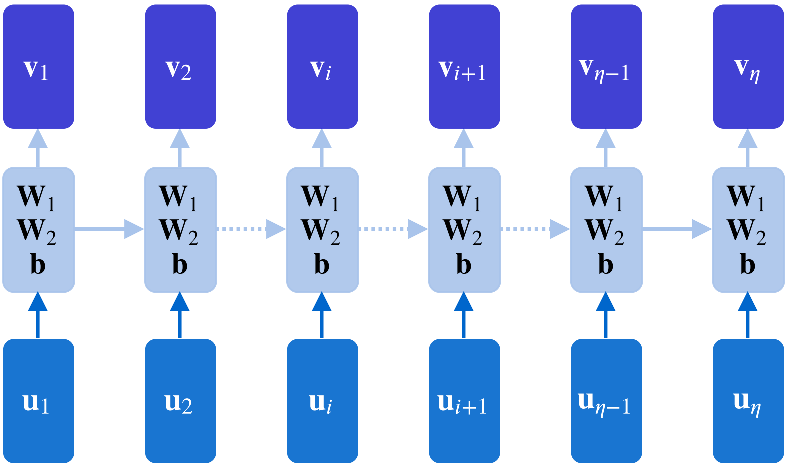


Figure 2.10: A recurrent layer in an RNN.

Stellenbosch University https://scholar.sun.ac.za

inner structure sequence-based specialisation [44]. An RNN consists of one or multiple *recurrent layers* where each layer shares its weights across each of the η inputs in a sequence $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_\eta\}$ as illustrated in Figure 2.10. For some time-sequence like a speech segment $\mathbf{u}$ consisting of η frames $\mathbf{u}_i$ over time, an RNN layer shares its weights across each frame. Each recurrent layer has trainable weight parameters $\mathbf{W}_1 \in \mathbb{R}^{(N\times N)}$, $\mathbf{W}_2 \in \mathbb{R}^{(N\times M)}$ and $\mathbf{b} \in \mathbb{R}^{(N\times 1)}$. By using $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{b}$, the recurrent layer produces an output $\mathbf{v}_i \in \mathbb{R}^{(N\times 1)}$ for each input $\mathbf{u}_i \in \mathbb{R}^{(M\times 1)}$ to the layer.

The layer output $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\eta\}$ is calculated from the layer input $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_\eta\}$ by using the following standard sequence-based specialisation function:

$$\mathbf{v}_i = \sigma(\mathbf{W}_1\mathbf{v}_{i-1} \ + \mathbf{W}_2\mathbf{u}_i + \mathbf{b}), \tag{2.4}$$

where σ is some activation function. $\mathbf{W}_1$ filters the sequence history $\mathbf{v}_{i-1}$ up until time step $i-1$ that is relevant to the input $\mathbf{u}_i$ to its output $\mathbf{v}_i$. At the same time $\mathbf{W}_2$ filters the relevant information in $\mathbf{u}_i$ to $\mathbf{v}_i$. These weights can be seen as a method that controls the flow of information from the previous $(i-1)$ and current (i) time steps to the layer output $\mathbf{v}$.

There are various sequence-based specialised configurations for RNNs that uses different gates to filter and scale the importance of certain information to the layer output. An LSTM adds an input, output and forget gate to the standard RNN [45] which enables the LSTM to take long- and short-term dependencies into account. The standard RNN cannot successfully handle long-term dependencies due to vanishing/exploding gradients [46]. A gated recurrent unit (GRU) which is an improvement on the LSTM [46, 47], adds a reset and update gate to the standard approach [45]. For all RNNs throughout this thesis we use GRUs and give the output dimension N for a recurrent GRU layer in the following format: (N).

Multiple recurrent layers can be connected by feeding the outputs of one layer $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\eta\}$, in order, as the inputs $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_\eta\}$ of the next layer. By using one or multiple connected recurrent layers, we can construct an RNN function $g_{\mathbf{\Phi}_{\text{RNN}}}(\mathbf{x})$ where $\mathbf{\Phi}$ is all the network's trainable parameters which includes the trainable weights $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{b}$ of each recurrent layer. The first recurrent layer's input is the input to the network $\mathbf{x}$ and the last recurrent layer's output is the output of the network $\hat{\mathbf{y}}$.

2.3.4 Classifiers

A classifier maps an input $\mathbf{x}^{(i)}$ to a category or class [44], thereby constructing a model function $f_{\mathbf{\Theta}}(\mathbf{x})$ between the input $\mathbf{x}^{(i)}$ and its predicted class output $\hat{\mathbf{y}}^{(i)}$ where $\mathbf{\Theta}$ is all the trainable model parameters. The general structure of a classifier is shown in Figure 2.11 where $f_{\boldsymbol{\theta}}(\mathbf{x})$ can be any one of the neural networks above (FFNN, CNN or RNN).

The output vector $\hat{\mathbf{y}}^{(i)}$ has dimension N where N is equal to the number of possible

Stellenbosch University https://scholar.sun.ac.za

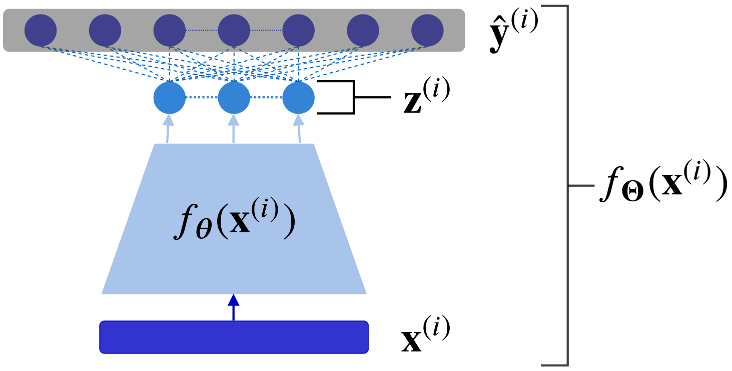

$\hat{\mathbf{y}}^{(i)}$
$\mathbf{z}^{(i)}$
$f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$
$f_{\boldsymbol{\Theta}}(\mathbf{x}^{(i)})$
$\mathbf{x}^{(i)}$

Figure 2.11: The general shape and structure of a classifier.

different classes. We use a softmax layer to produce $\hat{\mathbf{y}}^{(i)}$ thereby ensuring that $\hat{\mathbf{y}}^{(i)}$ contains the probabilities that a given input $\mathbf{x}^{(i)}$ belongs to each of the N classes. A softmax layer takes an input vector $\mathbf{z}^{(i)}$ and turns it into a probability distribution. Therefore all the values in $\hat{\mathbf{y}}^{(i)}$ sums to one. At test time the class with the highest probability is taken as an input's predicted class.

To train the classifier we use the multiclass log loss between the one-hot actual label vector $\mathbf{y}^{(i)}$ and the predicted class probabilities $\hat{\mathbf{y}}^{(i)}$ for a single training instance:

$$\begin{aligned}\ell_{\text{classifier}}(\,\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\,) &= -\sum_{c=1}^{N} \mathrm{y}_c^{(i)} \,\log(\,\hat{\mathrm{y}}_c^{(i)}\,) \\ &= -\sum_{c=1}^{N} \mathrm{y}_c^{(i)} \,\log(\,f_{\boldsymbol{\Theta}}(\mathbf{x}^{(i)})_c\,),\end{aligned} \tag{2.5}$$

where c simultaneously steps through the corresponding class values in $\mathbf{y}^{(i)}$ and $\hat{\mathbf{y}}^{(i)}$.

In some cases we might want to train a classifier on one dataset or domain and use it as a feature extractor for another. In these cases we can use the embedding $\mathbf{z}^{(i)}$ calculated by $f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$ as shown in Figure 2.11, as a latent feature representation for an input $\mathbf{x}^{(i)}$.

2.3.5 Siamese Neural Networks

A Siamese network does not classify an input, but measures the similarity between inputs [48, 49]. The network consists of identical subnetworks $f_{\boldsymbol{\Theta}}(\mathbf{x})$ with shared trainable model parameters $\boldsymbol{\Theta}$ as illustrated in Figure 2.12. Again, any of the networks above (FFNN, CNN or RNN) can be used as the basis $f_{\boldsymbol{\Theta}}(\mathbf{x})$ of the Siamese neural networks. A given input $\mathbf{x}^{(i)}$ is encoded by the subnetwork $f_{\boldsymbol{\Theta}}(\mathbf{x})$ to its feature embedding $\mathbf{z}^{(i)}$.

Ideally, inputs $\mathbf{x}_1^{(i)}$ and $\mathbf{x}_2^{(i)}$ of the same class should have similar feature embeddings $\mathbf{z}_1^{(i)}$ and $\mathbf{z}_2^{(i)}$, and inputs $\mathbf{x}_1^{(i)}$ and $\mathbf{x}_2^{(i)}$ of different classes should have different feature embeddings $\mathbf{z}_1^{(i)}$ and $\mathbf{z}_2^{(i)}$. Many studies [50–52] argued that this *relative* distance rather than an absolute distance is more promising. To learn this relative distance metric, we will use the triplet hinge loss to train the Siamese network.

Stellenbosch University https://scholar.sun.ac.za

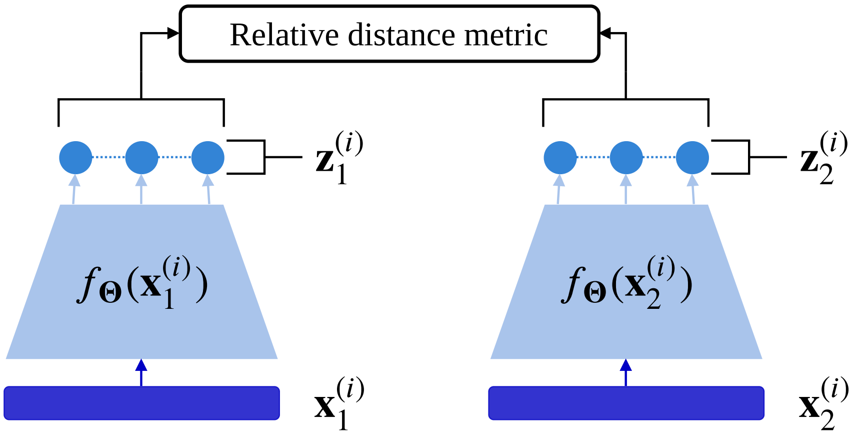
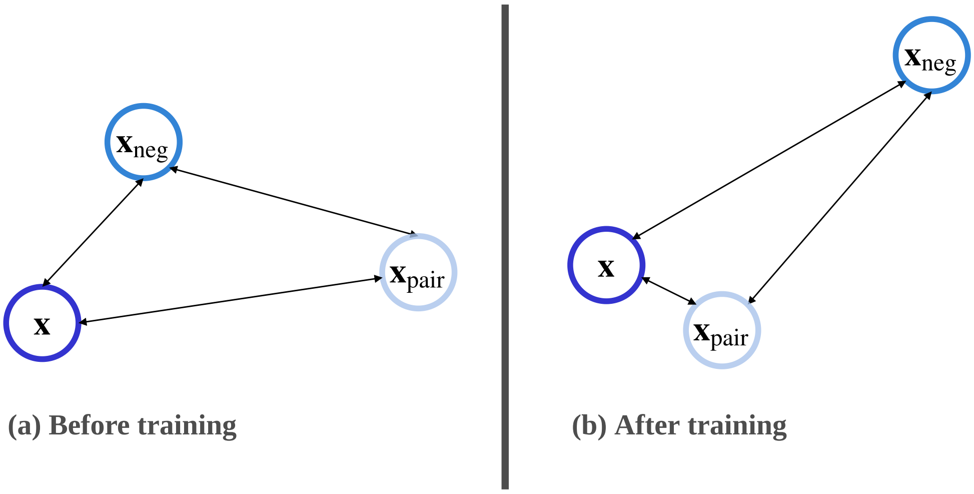


Figure 2.12: The Siamese neural network consists of two subnetworks with shared parameters.

A triplet hinge loss has inputs $\mathbf{x}$, $\mathbf{x}_{\text{pair}}$ and $\mathbf{x}_{\text{neg}}$. Specifically, $\mathbf{x}_{\text{pair}}$ is the positive anchor of $\mathbf{x}$ (i.e. $\mathbf{x}$ and $\mathbf{x}_{\text{pair}}$ are from the same class) and $\mathbf{x}_{\text{neg}}$ is the negative anchor of $\mathbf{x}$ (i.e. $\mathbf{x}$ and $\mathbf{x}_{\text{neg}}$ are from different classes). The triplet hinge loss aims to push the embeddings of $\mathbf{x}$ and $\mathbf{x}_{\text{pair}}$ closer whilst pushing the embeddings of $\mathbf{x}$ and $\mathbf{x}_{\text{neg}}$ further away from one another [50–54] with regards to some margin parameter m as illustrated in Figure 2.13. Logically, to do this the distance between the embeddings of $\mathbf{x}$ and $\mathbf{x}_{\text{pair}}$ should be smaller than the distance between $\mathbf{x}$ and $\mathbf{x}_{\text{neg}}$. The triplet hinge loss is given by:

$$l_{\text{triplet}}(\mathbf{x}, \mathbf{x}_{\text{pair}}, \mathbf{x}_{\text{neg}}) = \max\{0, m + d(\mathbf{x}, \mathbf{x}_{\text{pair}}) - d(\mathbf{x}, \mathbf{x}_{\text{neg}})\}, \tag{2.6}$$


$\mathbf{x}_{\text{neg}}$
$\mathbf{x}_{\text{pair}}$
$\mathbf{x}$
$\mathbf{x}_{\text{neg}}$
$\mathbf{x}$
$\mathbf{x}_{\text{pair}}$

(a) Before training

(b) After training

Figure 2.13: The aim of a triplet loss is to push the representations of inputs of the same class together and the representations of inputs from different classes away from each other.

Stellenbosch University https://scholar.sun.ac.za

where

$$
\begin{aligned}
d(\mathbf{x}_1, \mathbf{x}_2) &= \left\| \mathbf{z}_1 - \mathbf{z}_2 \right\|_2^2 \\
&= \left\| f_{\boldsymbol{\Theta}}(\mathbf{x}_1) - f_{\boldsymbol{\Theta}}(\mathbf{x}_2) \right\|_2^2
\end{aligned}
\tag{2.7}
$$

is the squared Euclidean distance between the embeddings $\mathbf{z}_1$ and $\mathbf{z}_2$ corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ and m is some margin parameter [50, 55].

2.3.6 Autoencoders

An autoencoder (AE) is an unsupervised neural network which aims to reconstruct its input through a lower dimensional latent representation that acts as an information bottleneck [56]. The bottleneck representation limits the amount of information that flows to the output of the network $\hat{\mathbf{y}}^{(i)}$. Therefore, the AE is forced to only capture the information necessary to reconstruct the input $\mathbf{x}^{(i)}$ at its output $\hat{\mathbf{y}}^{(i)}$.

As shown in Figure 2.14, the AE's encoder $f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$ encodes the input $\mathbf{x}^{(i)}$ to the latent embedding $\mathbf{z}^{(i)}$. The embedding $\mathbf{z}^{(i)}$ is used as a latent feature representation for a given input $\mathbf{x}^{(i)}$. The decoder $f_{\boldsymbol{\phi}}(\mathbf{z}^{(i)})$ decodes $\mathbf{z}^{(i)}$ to produce the network's output $\hat{\mathbf{y}}^{(i)}$.

The model function $f_{\boldsymbol{\Theta}}(\mathbf{x})$ is the combination of $f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$ and $f_{\boldsymbol{\Phi}}(\mathbf{z}^{(i)})$, where $\boldsymbol{\Theta}$ is all the trainable network parameters $\boldsymbol{\theta}$ of the encoder network $f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$ and $\boldsymbol{\phi}$ of the decoder network $f_{\boldsymbol{\Phi}}(\mathbf{z}^{(i)})$. To learn $f_{\boldsymbol{\Theta}}(\mathbf{x})$, we use a squared loss,

$$
\begin{aligned}
\ell(\,\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\,) &= ||\mathbf{y}^{(i)} - f_{\boldsymbol{\Theta}}(\mathbf{x}^{(i)})||_2^2 \\
&= ||\mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)}||_2^2,
\end{aligned}
\tag{2.8}
$$

between the network's output $\hat{\mathbf{y}}^{(i)}$ and the desired output $\mathbf{y}^{(i)}$. For the AE-loss, the target

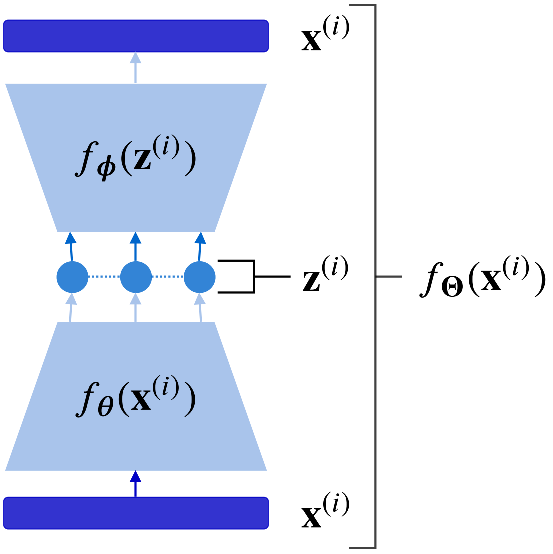

$\mathbf{x}^{(i)}$
$f_{\boldsymbol{\phi}}(\mathbf{z}^{(i)})$
$\mathbf{z}^{(i)}$
$f_{\boldsymbol{\Theta}}(\mathbf{x}^{(i)})$
$f_{\theta}(\mathbf{x}^{(i)})$
$\mathbf{x}^{(i)}$

Figure 2.14: The general structure and parts of an AE.

Stellenbosch University https://scholar.sun.ac.za

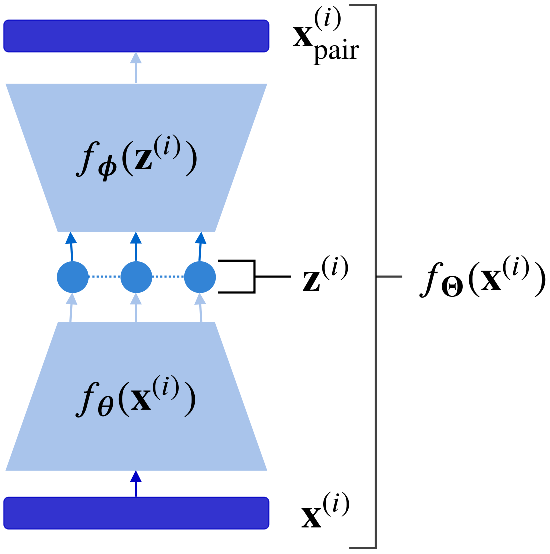

$\mathbf{x}^{(i)}_{\text{pair}}$
$f_{\phi}(\mathbf{z}^{(i)})$
$\mathbf{z}^{(i)}$
$f_{\Theta}(\mathbf{x}^{(i)})$
$f_{\theta}(\mathbf{x}^{(i)})$
$\mathbf{x}^{(i)}$

Figure 2.15: The general shape and structure of a CAE.

$\mathbf{y}^{(i)}$ is set to $\mathbf{x}^{(i)}$ in Equation 2.8:

$$\begin{aligned}\ell_{\text{AE}}\big(\,\mathbf{x}^{(i)},\mathbf{x}^{(i)}\,\big) &= ||\mathbf{x}^{(i)} - f_{\mathbf{\Theta}}(\mathbf{x}^{(i)})||_2^2 \\ &= ||\mathbf{x}^{(i)} - \hat{\mathbf{y}}^{(i)}||_2^2.\end{aligned} \tag{2.9}$$

2.3.7 Correspondence Autoencoders

The correspondence autoencoder (CAE) is identical to the AE with its encoder-decoder structure as shown in Figure 2.15. However, instead of reproducing the input $\mathbf{x}^{(i)}$ at its output, it aims to produce another instance of the same class as the input $\mathbf{x}^{(i)}_{\text{pair}}$ (a pair of the input) [21]. For the CAE-loss, we therefore set the target $\mathbf{y}^{(i)}$ to $\mathbf{x}^{(i)}_{\text{pair}}$ in Equation 2.8:

$$\begin{aligned}\ell_{\text{CAE}}\big(\,\mathbf{x}^{(i)},\mathbf{x}^{(i)}_{\text{pair}}\,\big) &= ||\mathbf{x}^{(i)}_{\text{pair}} - f_{\mathbf{\Theta}}(\mathbf{x}^{(i)})||_2^2 \\ &= ||\mathbf{x}^{(i)}_{\text{pair}} - \hat{\mathbf{y}}^{(i)}||_2^2.\end{aligned} \tag{2.10}$$

Again, we use the embedding $\mathbf{z}^{(i)}$ produced by the encoder $f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$ as the latent feature representation for a given input $\mathbf{x}^{(i)}$.

2.3.8 Optimisation of the Loss

To train neural networks, we use gradient decent to optimise the objective function $\ell\big(\,\mathbf{X}^{(i)},\mathbf{Y}^{(i)}\,\big)$ of the model $f_{\mathbf{\Theta}}(\mathbf{X}^{(i)})$ under consideration. For a singular training example all the models inputs are grouped under $\mathbf{X}^{(i)}$, all its outputs are grouped under $\hat{\mathbf{Y}}^{(i)}$ and all its desired outputs are grouped under under $\mathbf{Y}^{(i)}$. Our optimisation strategy is to minimise the function $\ell\big(\,\mathbf{X}^{(i)},\mathbf{Y}^{(i)}\,\big)$ of the model $f_{\mathbf{\Theta}}(\mathbf{X}^{(i)})$ by calculating the derivative of the function with respect to $\mathbf{X}^{(i)}$ to obtain the function's gradient $\ell'\big(\,\mathbf{X}^{(i)},\mathbf{Y}^{(i)}\,\big)$. We use

Stellenbosch University https://scholar.sun.ac.za

this gradient $\ell'\big(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\big)$ to update the trainable parameters $\boldsymbol{\Theta}$ to improve the values of the network outputs $\hat{\mathbf{Y}}^{(i)}$ to be closer to its desired outputs $\mathbf{Y}^{(i)}$ [44].

In order to adapt the $\boldsymbol{\Theta}$ parameters for a single training example consisting of $\mathbf{X}^{(i)}$ and $\mathbf{Y}^{(i)}$, we do a forward pass to get the value of $\ell\big(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\big)$. Thereafter we do a backward pass to get $\ell'\big(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\big)$ with respect to $\mathbf{X}^{(i)}$. This process of calculating the model's gradients is known as backpropagation. Finally, we use the gradient $\ell'\big(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\big)$ to update $\boldsymbol{\Theta}$ according to the following function:

$$\boldsymbol{\Theta} = \boldsymbol{\Theta} - \beta\, \ell'\big(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\big), \tag{2.11}$$

where the learning rate β scales the amount of $\ell'\big(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\big)$ used to update $\boldsymbol{\Theta}$.

This entire process of using backpropagation on a single training example to update the training parameters $\boldsymbol{\Theta}$ is known as stochastic gradient decent (SGD). However, instead of doing this for a single example at a time, we use the sum of $\ell\big(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\big)$ over τ number of training examples in a batch to update $\boldsymbol{\Theta}$:

$$\boldsymbol{\Theta} = \boldsymbol{\Theta} - \beta \left(\frac{1}{\tau} \sum_{i=1}^{\tau} \ell'\big(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\big) \right). \tag{2.12}$$

This is known as minibatch SGD. Essentially, this means we update the parameters $\boldsymbol{\Theta}$ in a direction that results in a reduction of the loss with each subsequent batch of training examples.

There are several extensions of SGD which improves model training and convergence. We specifically use Adam optimisation [57], which incorporates an adaptive learning rate so that each of the parameters in $\boldsymbol{\Theta}$ are updated with its own local learning rate parameter.

2.4 Implementation and Resources

All neural networks we consider throughout the thesis are implemented in TensorFlow [58] and Python. Each network we consider is trained using backpropagation (Chapter 2.3.8), specifically using Adam optimisation [57], with a learning rate of 10^{-3}. We use an Nvidia GEFORCE RTX 2070 GPU with 8Gb memory to train our models.

In Chapters 3 to 5, we consider various models to embed spoken words and images to a representation embedding $\mathbf{z}$. We always use an embedding dimensionality of 130 for all these representation embeddings. This is to ensure all the results across Chapters 3 to 5 are comparable.

Stellenbosch University https://scholar.sun.ac.za

2.5 Chapter Summary

This chapter explained unimodal few-shot learning and discussed the approaches followed by relevant studies in this field. We also gave general explanations of the neural network building blocks that we will use to construct our models and the methods to train these models. The speech and image datasets explained in this chapter will be used throughout the rest of the thesis. In the next chapter, we look at the study done by Eloff et al. [1] on multimodal few-shot learning which is a multimodal expansion of unimodal few-shot learning. We devote the entire Chapter 3 to improve the experimental setup of Eloff et al. [1], thereby developing our own experimental setup.

Stellenbosch University https://scholar.sun.ac.za

Chapter 3

Multimodal Few-Shot Learning using Transfer Learning

This chapter thoroughly investigates and re-implements the multimodal few-shot learning models proposed by Eloff et al. [1], which specifically investigated multimodal few-shot learning of digit words and images. We start this chapter by introducing the multimodal speech-to-image matching task. After this, in Chapter 3.2 we briefly outline work that inspired this task before explaining the various speech-vision few-shot models developed by [1] in Chapter 3.3 and 3.4.

The goal of this chapter is to re-implement the models of Eloff et al. [1] as the baseline for our multimodal (speech-vision) few-shot learning work in Chapters 4 and 5. We do this for three reasons: (1) Eloff et al. [1] is the first study to consider multimodal few-shot learning, (2) we also consider multimodal few-shot learning of spoken word and image digits, and (3) this allows us to base our experimental framework on the one used in [1].

3.1 Multimodal Few-Shot Matching

Multimodal few-shot matching is the task of matching corresponding unlabelled inputs from different modalities after being presented with only a multimodal support set [1].

> **Multimodal few-shot learning** is the task of learning a new concept from a few paired examples of this concept, where each pair consists of items from different modalities but of the same concept.

These few cross-modal paired examples are known as a *multimodal support set*. Any two modalities can be used for this matching task. We specifically use a speech-to-image few-shot matching task as explained in Chapter 3.1.1. The speech-to-image matching task is constructed from a unimodal speech classification task and a unimodal image classification task as explained in Chapter 2.1.1 [1]. As a first approach to do this task, we use an indirect matching approach discussed in Chapter 3.1.2.

Stellenbosch University https://scholar.sun.ac.za

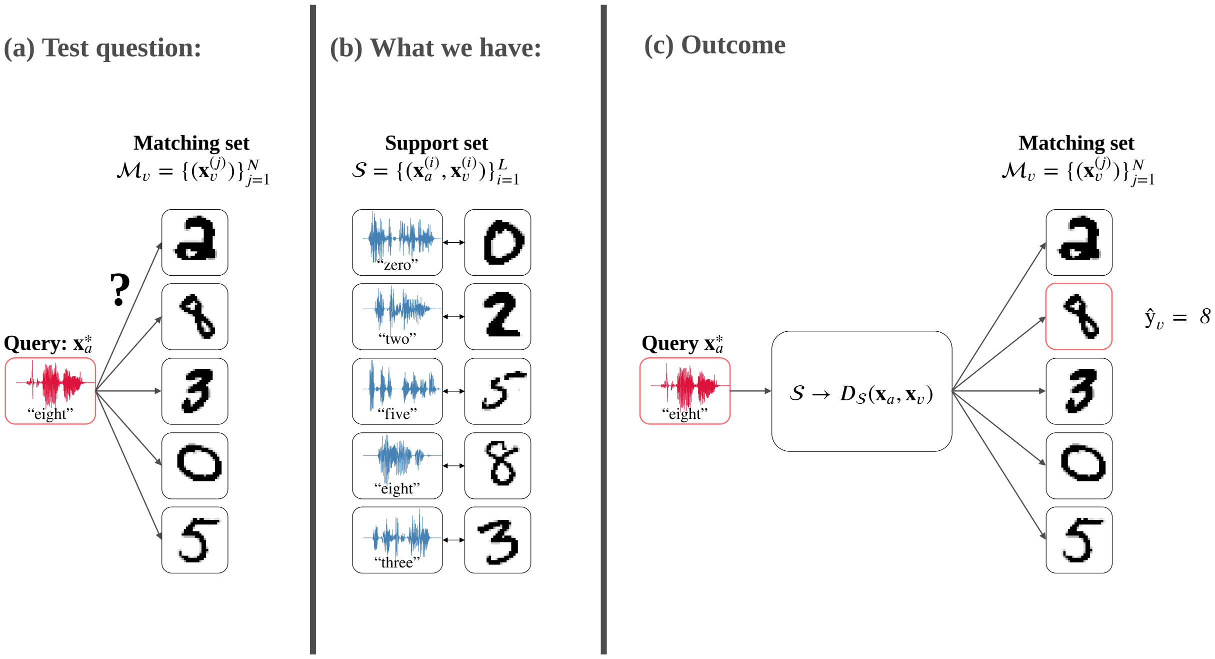

(a) Test question:
Matching set
$\mathcal{M}_v = \{(\mathbf{x}_v^{(j)})\}_{j=1}^N$
?
Query: $\mathbf{x}_a^*$
"eight"
(b) What we have:
Support set
$\mathcal{S} = \{(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})\}_{i=1}^L$
"zero"
"two"
"five"
"eight"
"three"
(c) Outcome
Matching set
$\mathcal{M}_v = \{(\mathbf{x}_v^{(j)})\}_{j=1}^N$
Query $\mathbf{x}_a^*$
"eight"
$\mathcal{S} \rightarrow D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$
$\hat{y}_v = 8$

Figure 3.1: Multimodal one-shot speech-to-image matching is portrayed by (a) the question shown at test time. By only using (b) the multimodal support set a model should find (c) a distance metric $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$ to solve this question.

3.1.1 The Multimodal Speech-to-Image Matching Task

At test time, a multimodal few-shot model is presented with an unseen unlabelled speech query $\mathbf{x}_a^*$ and prompted to identify the corresponding image of the same concept in a matching set $\mathcal{M}_v = \{(\mathbf{x}_v^{(j)})\}_{j=1}^N$ of unseen unlabelled test images as shown in Figure 3.1(a). To do this task, the model is only given a multimodal support set $\mathcal{S} = \{(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})\}_{i=1}^L$ that consists of a speech-image pair for each of the L classes, where each pair contains an isolated spoken word $\mathbf{x}_a^{(i)}$ and a corresponding image $\mathbf{x}_v^{(i)}$ of the same class as $\mathbf{x}_a^{(i)}$. Although Figure 3.1(b) illustrates a one-shot support set containing $L = 5$ classes, there are eleven possible digit classes which includes the classes "one" to "nine" as well as "oh" and "zero". The image instances of both the digit classes "oh" and "zero", are images of a *0*. For the *one-shot* case, the multimodal one-shot support set $\mathcal{S}$ illustrated in Figure 3.1(b), consists of *one* example pair for each of the L classes. However, for the few-shot or K-shot setting, a multimodal K-shot support set $\mathcal{S} = \{(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})\}_{i=1}^{L \times K}$ consists of K speech-image pairs for each of the L classes. Neither the speech query $\mathbf{x}_a^*$ nor the matching set items $\mathcal{M}_v$ occur exactly in the multimodal support set $\mathcal{S}$

As illustrated in Figure 3.1(c), to match unlabelled speech queries $\mathbf{x}_a$ to unlabelled matching images, we need some distance metric $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$ between speech instances $\mathbf{x}_a$ and image instances $\mathbf{x}_v$, i.e. for the K-shot setting we need a multimodal K-shot learning model. More specifically, a multimodal K-shot learning model learns $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$ from a multimodal K-shot support set $\mathcal{S}$. For the one-shot setting ($K = 1$) in Figure 3.1, $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$ is a multimodal one-shot learning model.

Stellenbosch University https://scholar.sun.ac.za

3.1.2 An Indirect Matching Approach

To perform the speech-to-image matching task, we use an indirect matching approach which consists of the multimodal support set $\mathcal{S}$ and two unimodal comparisons (speech-speech and image-image comparisons) as illustrated in Figure 3.2(a). This indirect approach forms a distance metric $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$ between audio queries and test images as shown in Figure 3.2(b). For the unimodal comparisons, we need separate unimodal speech and vision networks that can produce feature representations from which within-modality similarity can be measured. I.e. each unimodal model should produce similar feature representations for all instances of an unseen few-shot class.

More formally, this indirect approach works as follows. First, we use a unimodal speech network to extract speech representations $\mathbf{z}_a$ for the query $\mathbf{x}_a^*$ and each $\mathbf{x}_a^{(i)}$ in $\mathcal{S}$. We then compare the query representation $\mathbf{z}_a^*$ to each representation $\mathbf{z}_a^{(i)}$ in $\mathcal{S}$ to find the query's closest spoken neighbour in $\mathcal{S}$. A unimodal vision network is then used to extract image representations $\mathbf{z}_v$ for this closest neighbour's paired image $\mathbf{x}_v^{(i)}$ and each image $\mathbf{x}_v^{(j)}$ in the matching set $\mathcal{M}_v$. The representation of the paired image $\mathbf{z}_v^{(i)}$ is then compared to each image representation $\mathbf{z}_v^{(j)}$ in $\mathcal{M}_v$ to find the closest image neighbour of $\mathbf{z}_v^{(i)}$ in $\mathcal{M}_v$. This closest image is taken as the model's prediction for the query's matching image. In Figure 3.2(a), this is the image of the rightmost *eight*.

To evaluate the representations produced by the different unimodal models for the unimodal comparisons in the speech-to-image matching task, we implement this task with *multimodal episodes*. Each multimodal episode is an instance of a K-shot speech-to-image matching task which contains a query set, a K-shot multimodal support set $\mathcal{S}$ and a

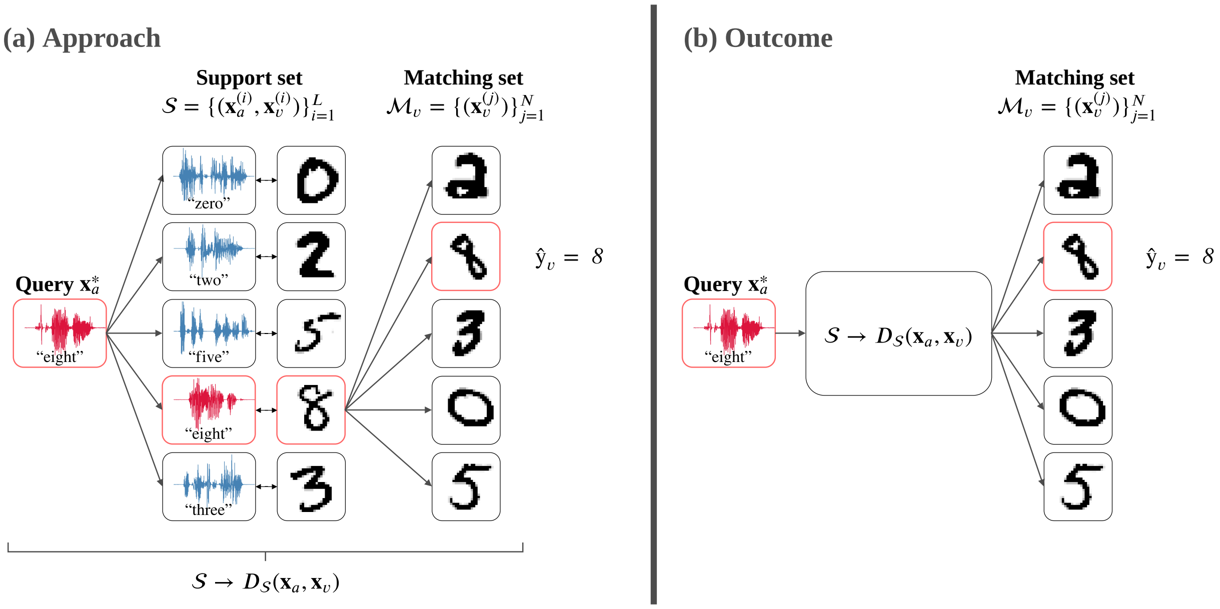

(a) Approach
Support set
$\mathcal{S} = \{(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})\}_{i=1}^{L}$
Matching set
$\mathcal{M}_v = \{(\mathbf{x}_v^{(j)})\}_{j=1}^{N}$
Query $\mathbf{x}_a^*$
“zero”
“two”
“five”
“eight”
“three”
“eight”
$\hat{y}_v = 8$
$\mathcal{S} \rightarrow D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$
(b) Outcome
Matching set
$\mathcal{M}_v = \{(\mathbf{x}_v^{(j)})\}_{j=1}^{N}$
Query $\mathbf{x}_a^*$
“eight”
$\mathcal{S} \rightarrow D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$
$\hat{y}_v = 8$

Figure 3.2: One approach to do multimodal one-shot speech-to-image matching is to (a) use speech-speech and image-image comparisons across the support set to obtain (b) a distance metric $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$.

Stellenbosch University https://scholar.sun.ac.za

matching set $\mathcal{M}_v$. For each episode, the episode's K-shot support set should be used to match each query in the episode's query set to a matching image in the episode's matching set.

Similarly as Vinyals et al. [8], Ravi and Larochelle [23] and Eloff et al. [1], in this chapter as well as in Chapter 4, we use cosine distance to compare within-modality feature representations. Various unimodal embedding models to produce representations for these speech-speech and image-image comparisons, are discussed in Chapter 3.3 and Chapter 3.4.

3.2 Related Work

Due to DL there has been an abundance of advances in speech and image recognition systems in recent years. However, these advances caused ML systems to become dependent on complex deep neural networks with millions of parameters [59, 60]. As a result of this increasing complexity, systems became heavily dependent on large datasets of labelled image data or transcribed speech audio [2]. Most of these DL solutions only perform well when the training and testing data domains are made up of the same distributions, i.e. if the training and testing data are from the same domain (we refer to this setting as "in-domain"). However, if the data distribution changes such as for instance the model needs to be updated to include new data classes, the model has to be retrained from scratch on a large amount of new labelled training data.

In practice, the recollection and labelling of data is expensive and time-consuming. Therefore it would be useful to obtain models that can transfer the knowledge learned from some labelled dataset to another unlabelled dataset. For example, Donahue et al. [61] trained a model called DeCAF on a set of object recognition tasks. DeCAF can then be applied to related but new tasks which has too few training data to properly train a model from scratch.

In recent years unimodal few-shot learning, explained in Chapter 2.1, have been investigated for the purpose of finding DL methods that can learn from few labelled data examples. This was vaguely inspired by a child's ability to learn rapidly in a weakly supervised environment: children can learn new words and objects from only a few word or object examples [12–15]. For example, after hearing a novel word once, a child can infer the likely meaning of the word [11]. Additionally, it is plausible that children use existing knowledge to learn new words and objects [11]. As a result transfer learning is the most common approach to do unimodal few-shot learning as discussed in Chapter 2.1.2.

Moreover, children are able to learn the relationship between a new spoken word and visual object from only a few paired examples [13]. This led Eloff et al. [1] to propose the idea of multimodal few-shot learning as a solution towards acquiring systems that can learn as rapidly with as limited supervision as children. In addition, multimodal few-shot

Stellenbosch University https://scholar.sun.ac.za

models have the added benefit of learning what the data representations of a class look like in two different modalities. This could lead the way to obtain multimodal engineering systems that are less data-dependant and more efficient and flexible.

Specifically, Eloff et al. [1] investigated the use of transfer learning for multimodal few-shot learning of isolated spoken digits paired with digit images. They construct a multimodal few-shot learning model from two unimodal models (a speech network and a vision network) and a multimodal support set as described in Chapter 3.1.2. For each unimodal model to learn a distance metric within its specific modality, they attempt to use transfer learning to find unimodal models which generalises to classes not seen during training [7, 51, 61]. I.e. without any training, the models can find similar representations for all the instances of the same unseen few-shot class.

For these speech and vision networks, Eloff et al. [1] trained supervised classifiers and Siamese triplet networks on background data not containing any of the few-shot test classes. A multimodal few-shot learning model therefore consists of separate unimodal speech and vision versions of these networks, e.g. a multimodal few-shot classifier consists of a speech classifier and an image classifier. They evaluated these models against a baseline which uses direct unimodal comparisons on raw speech and image data (Chapter 3.3).

3.3 Baseline: Dynamic Time Warping and Pixels

As a baseline we use a method that uses no DL in order to determine whether using DL solutions are necessary or beneficial to the speech-to-image matching task. To do this, the baseline uses unimodal comparisons directly on raw speech and image data.

Specifically for the image-image comparisons, we flatten each two-dimensional image to a vector of size 1×784. The cosine distance over these flattened image pixels can then be used to compare two images.

For the speech-speech comparisons, we use dynamic time warping (DTW) over the MFCCs of spoken words. DTW is a method which uses dynamic programming to find the optimal alignment between two vectorised time series (like speech utterances) of variable length [2]. We add first (delta) and second (double-delta) order derivatives to these MFCCs, since Kamper et al. [62] found that delta and double-delta features are beneficial for DTW. Throughout this thesis when we use DTW, we use cosine distance as the frame-level distance metric.

3.4 Unimodal Transfer Learning Models

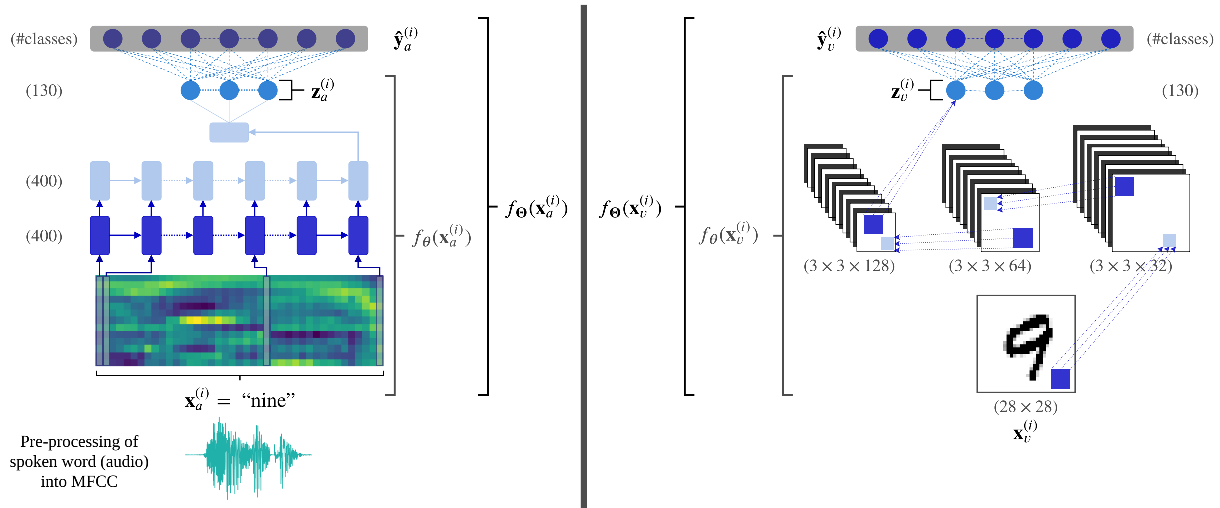
To get within-modality metrics for the speech-to-image matching task, we consider unimodal models that embeds all instances of a class to similar feature representations. Since the unimodal models do not see any of the few-shot classes during training, the unimodal

Stellenbosch University https://scholar.sun.ac.za

models should be able to generalise to such an extent that it would still produce similar representations for an unseen class [8].

We consider transfer learned classifiers in Chapter 3.4.1 and Siamese triplet networks in Chapter 3.4.2 for the unimodal models. More specifically for the speech models, we train a supervised classifier and a supervised Siamese triplet network on labelled background speech data. Similarly for the vision models, we train a supervised classifier and a supervised Siamese triplet network on labelled background image data. The background data does not contain any instances of the target few-shot classes seen at test time. The hope is that features learned on the background data would transfer to the unseen classes at test time.

A multimodal few-shot model consists of corresponding speech and vision networks, e.g a multimodal few-shot Siamese triplet model consists of a speech Siamese triplet network and a vision Siamese triplet network. To do unimodal comparisons at test time, we extract representations for each instance seen in the speech-to-image matching task from a multimodal few-shot model's unimodal networks. For each word instance, we do a forward pass through a multimodal few-shot model's speech network and extract the $\mathbf{z}_a$ embedding layer as the feature representation for the word instance. Similarly, for a given image instance, we do a forward pass through a multimodal few-shot model's vision network and extract the $\mathbf{z}_v$ embedding layer as the feature representation for the image.


(#classes)
$\hat{\mathbf{y}}_a^{(i)}$
(130)
$\mathbf{z}_a^{(i)}$
(400)
(400)
$f_\theta(\mathbf{x}_a^{(i)})$
$f_\Theta(\mathbf{x}_a^{(i)})$
$\mathbf{x}_a^{(i)}$ = "nine"
Pre-processing of spoken word (audio) into MFCC
$f_\Theta(\mathbf{x}_v^{(i)})$
$f_\theta(\mathbf{x}_v^{(i)})$
$\hat{\mathbf{y}}_v^{(i)}$
(#classes)
$\mathbf{z}_v^{(i)}$
(130)
(3 × 3 × 128)
(3 × 3 × 64)
(3 × 3 × 32)
(28 × 28)
$\mathbf{x}_v^{(i)}$

(a) Speech network

(b) Vision network

Figure 3.3: The multimodal classifier consists of (a) a speech classifier and (b) a vision classifier. (a) A speech classifier RNN is used to learn feature representations for speech data and (b) a vision classifier CNN is used to learn feature representations for image data.

Stellenbosch University https://scholar.sun.ac.za

3.4.1 Classifiers

To construct a multimodal classifier model, we train separate speech and vision classifiers (Chapter 2.3.4). The architectures for the speech and vision classifiers we use, are shown in Figure 3.3. Both these transfer learned classifier networks are trained with the multiclass log loss in Equation 2.5 on background labelled data that does not contain any of the few-shot classes seen at test time.

For the speech model $f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)})$ shown in Figure 3.3(a), we use an RNN followed by two fully connected layers, a latent representation layer $\mathbf{z}_a^{(i)}$ and then a softmax layer $\hat{\mathbf{y}}_a^{(i)}$. The vision model $f_{\mathbf{\Theta}}(\mathbf{x}_v^{(i)})$ shown in Figure 3.3(b) uses a CNN followed by a fully connected latent representation layer $\mathbf{z}_v^{(i)}$ and then a fully connected softmax layer $\hat{\mathbf{y}}_v^{(i)}$. We consider classifiers since, when given an input $\mathbf{x}^{(i)}$, the feature embedding $\mathbf{z}^{(i)}$ before the softmax layer should contain the necessary information to predict the correct class of $\mathbf{x}^{(i)}$. The hope is that this knowledge would transfer to classes not seen during training.

3.4.2 Siamese Triplet Networks

Considering that the feature representations play a major role in the speech-to-image matching task, we consider Siamese triplet networks. As discussed in Chapter 2.3.5, the triplet hinge loss used to train a Siamese network, forces the network to learn a relative distance metric between the feature representations from various classes: representations from the same class should ideally be closer to one another than representations from different classes.

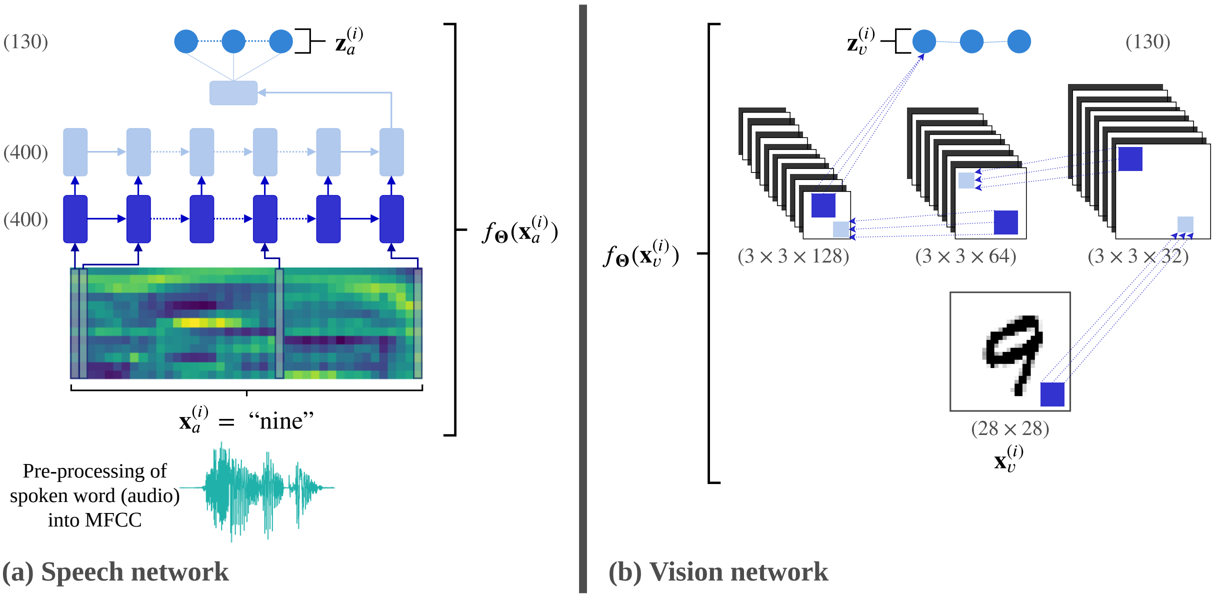


Figure 3.4: The multimodal Siamese triplet model consists of (a) a speech Siamese triplet network and (b) a vision Siamese triplet network. (a) A speech Siamese triplet RNN is used to learn feature representations for speech data and (b) a vision Siamese triplet CNN is used to learn feature representations for image data.

Stellenbosch University https://scholar.sun.ac.za

A speech Siamese triplet network and a vision Siamese triplet network are trained on background labelled data not containing any of the few-shot classes seen during testing. The hope is that the Siamese networks will capture a relative distance metric that can be transferred to distinguish between classes not seen during training. The multimodal Siamese triplet model consists of this speech and vision Siamese triplet networks.

For the speech Siamese triplet network, we use an RNN followed by a fully connected feature representation layer $\mathbf{z}_a^{(i)}$ with the specific architecture we use for this speech network shown in Figure 3.4(a). The vision Siamese triplet network consists of a CNN followed by a fully connected feature representation layer $\mathbf{z}_v^{(i)}$ where Figure 3.4(b) shows the exact architecture we use for this network.

For each of these speech and vision Siamese triplet networks, a single training instance consists of $\mathbf{x}$, $\mathbf{x}_{\text{pair}}$ and $\mathbf{x}_{\text{neg}}$. Similarly to Eloff et al. [1], we train the Siamese networks on mini-batches using the *batch all* strategy. Hermans et al. [54] showed that this strategy leads to better performance of the triplet network by pushing hard positive pairs closer and hard negative pairs further away. Each mini-batch samples p classes with k examples per class so that each mini-batch contains pk hard positive pairs $(\mathbf{x}, \mathbf{x}_{\text{pair}})$. Therefore, for each of the p sampled classes, the mini-batch consists of every possible pair that can be made up from the sampled class and each of its k examples. To sample negative items, we use the online semi-hard mining scheme: for each one of the pk positive pairs $(\mathbf{x}, \mathbf{x}_{\text{pair}})$, we find the most difficult negative pair $(\mathbf{x}, \mathbf{x}_{\text{neg}})$ according to some constraints [52–54].

3.5 Experimental Setup

In Chapter 3.5.1 we discuss the implementation of the different multimodal few-shot learning models (Chapter 3.4) and then evaluate these models on three possible tasks discussed in Chapter 3.5.2.

3.5.1 Models

We train a supervised speech classifier RNN and a supervised speech Siamese triplet RNN on labelled isolated words from the background Buckeye training set (Chapter 2.2.1.2) which does not contain any of the few-shot digit classes seen at test time. To perform early stopping, the speech models are validated on unimodal one-shot speech classification using the Buckeye validation subset. The hyperparameters of the speech networks are tuned on a unimodal one-shot speech classification task using the Buckeye test subset.

Similarly, we train a supervised vision classifier CNN and a supervised vision Siamese triplet CNN on the background Omniglot character images (Chapter 2.2.2.2) that does not contain any of the few-shot digit image classes seen during testing. The vision models are validated on unimodal one-shot image classification using the validation subset of

Stellenbosch University https://scholar.sun.ac.za

the Omniglot character images in order to perform early stopping. The vision networks' hyperparameters are tuned on a unimodal one-shot image classification task using the Omniglot test subset. All unimodal (speech and vision) models in this thesis are validated and tuned in the same manner as set out here.

The speech classifier with architecture shown in Figure 3.3(a) is trained with a batch size of 512. Figure 3.3(b) shows the architecture we use for the vision classifier trained with a batch size of 64. For both the speech and vision Siamese triplet networks, we use $k = 8$ and $p = 88$ since 88 is the maximum number of sampling classes we could fit on a single GPU. Furthermore, we use $m = 0.7$ for the triplet margin (Chapter 2.3.5) and we perform L_2 normalisation on the feature representations before using them either in training, validation or testing. The speech Siamese network with architecture shown in Figure 3.4(a) is trained on 175 of the mini-batches described in Chapter 3.4.2. The vision Siamese network is trained on 200 of these mini-batches with an architecture as shown in Figure 3.4(b).

As described in Chapter 3.4, the multimodal few-shot learning models consists of corresponding speech and vision networks, e.g. a multimodal classifier consists of a speech classifier and a vision classifier.

3.5.2 Evaluation Setup

The multimodal few-shot classifier and Siamese models are evaluated on a multimodal speech-to-image matching task, with the implementation of this task given in Chapter 3.5.2.1. In order to investigate the performance of a multimodal few-shot learning model's different parts, we can also evaluate their separate speech and vision networks. The speech classifier and Siamese RNNs are evaluated on a unimodal speech classification task, with the implementation of this task given in Chapter 3.5.2.2. Similarly, the vision classifier and Siamese CNNs are evaluated on a unimodal image classification task, with the implementation of this task given in Chapter 3.5.2.3. All unimodal and multimodal one- and five-shot experiments are done on the MNIST and TIDigits test subsets. We report multimodal and unimodal accuracies with 95% confidence intervals averaged over five models trained with different seeds.

3.5.2.1 Multimodal Speech-to-Image Matching Task Implementation

Chapter 3.1.2 discussed an indirect approach to do the multimodal speech-to-image matching task. In the implementation of this task, each accuracy score is an average over 400 multimodal episodes sampled from the MNIST and TIDigits test subsets.

In each multimodal episode, a multimodal K-shot support set is constructed by randomly sampling K spoken digit and image digit pairs for each of the $L = 11$ classes ("one" to "nine", as well as "zero" and "oh"). For the episode's matching set, we sample

Stellenbosch University https://scholar.sun.ac.za

ten digit images not in the support set. The matching set only contains ten digit images since there are only ten unique handwritten digit classes. Therefore, if the speech query is either a "zero" or an "oh", it is counted as correct if the model's prediction of a matching image is that of a 0. Finally, within an episode, ten different speech query instances (also not in the support set) are also sampled while keeping the support and matching sets fixed. Each speech query has to be matched to the correct image in the matching set.

3.5.2.2 Speech Classification Task Implementation

The unimodal speech classification task is discussed in Chapter 2.1.1. Each reported accuracy score is an average over 400 unimodal speech classification episodes sampled from the TIDigits test subset. The speech classification episodes are sampled similarly to the multimodal episodes above, but a matching set is not sampled and instead of a multimodal K-shot support set, we sample a unimodal K-shot speech support. The unimodal K-shot speech support set consists of K sampled spoken digits paired with their class labels for each of the $L = 11$ classes. Within an episode, each of the ten sampled speech query instances (which is not in the speech support set) should then be matched to the correct label in the support set.

3.5.2.3 Image Classification Task Implementation

The unimodal image classification task is discussed in Chapter 2.1.1. Each unimodal image classification accuracy score is an average over 400 unimodal image episodes sampled from the MNIST test subset. The unimodal image classification episodes are sampled similarly to the speech classification episodes above. However, instead of sampling ten speech queries and a unimodal K-shot speech support set, we sample ten image query instances and a unimodal image K-shot support set. For each of the $L = 10$ classes (*0* to *9* since there are only ten unique image digit classes), the unimodal image K-shot support set consists of K digit images paired with their class labels. Within each episode, each of the sampled image queries (which is not in the image support set) should be matched to its correct label in the image support set.

3.6 Experiments

Ultimately we want to test the multimodal models on the speech-to-image matching task (Chapter 3.6.3). To get to this, we start with developmental experiments which investigates Eloff et al.'s [1] experiments by implementing our model architectures and theirs in our experimental environment. Before getting to the speech-to-image matching task, we first consider the unimodal speech and vision networks that the multimodal few-shot models

Stellenbosch University https://scholar.sun.ac.za

Table 3.1: The image classification results produced with Eloff et al.'s [1] code.

Model		10-way accuracy (%)	
		one-shot	five-shot
Classifier CNN [1]	Before mistake fix	64.19 ± 0.70	**85.11 ± 0.34**
	After mistake fix	**64.23 ± 0.70**	85.04 ± 0.35
Siamese CNN [1]	Before mistake fix	67.23 ± 0.86	86.58 ± 0.45
	After mistake fix	**69.46 ± 1.31**	**87.75 ± 0.69**

comprises of, in isolation. To do this we evaluate the speech networks on a unimodal speech classification task and the vision networks on a unimodal image classification task.

3.6.1 Comparing Our Results to Eloff et al.'s

Our speech models are not directly comparable to Eloff et al.'s [1] since we use RNNs instead of CNNs. RNNs are more appropriate than CNNs to model variable length spoken words (Chapter 2.3.3).

In contrast to the speech models, our vision models should be directly comparable to Eloff et al.'s [1]. Their image classification results could not be reproduced by the code they published. In addition a mistake in Eloff et al.'s [1] validation setup was found: they trained and validated on exactly the same data. Table 3.1 shows the actual results produced by their code before and after the mistake was fixed. We will compare our results to the actual results produced by their code after the mistake was fixed.

In Chapter 3.6.1.1 we specifically consider whether our implementation is comparable to Eloff et al.'s [1]. Since we use less sampling classes p than Eloff et al. [1], we investigate in Chapter 3.6.1.2 whether this affects the results.

3.6.1.1 Our Implementation vs. Eloff et al.'s Implementation

We use a different architecture for the vision classifier and Siamese networks than Eloff et al. [1] to ensure comparability across all models in Chapters 3, 4 and 5. In our environment, we implement our architecture and Eloff et al.'s [1] architecture. Table 3.2 compares these two architectures when used for the classifier, and Table 3.3 compares these architectures when used for the Siamese network.

From Table 3.2 we can see the accuracy scores for different classifier architectures are sufficiently similar. Our classifier scores are only slightly worse than Eloff et al.'s [1] implementation of their architecture (Classifier after mistake fix). Although our classifier architecture performs well enough in comparison to theirs, the same does not hold for our Siamese architecture.

Stellenbosch University https://scholar.sun.ac.za

Table 3.2: Our implementation of the classifier vision models with our architecture vs. Eloff et al.'s [1] architecture.

	Architecture	10-way accuracy (%)	
		one-shot	five-shot
Their implementation	Classifier after mistake fix (from Table 3.1, row 2)	64.23 ± 0.70	85.04 ± 0.35
Our implementation	Our architecture	**63.23 ± 1.42**	**82.90 ± 1.12**
	Eloff et al.'s [1] architecture	61.96 ± 2.21	81.42 ± 1.48

Table 3.3 shows our Siamese architecture and Eloff et al.'s [1] Siamese architecture trained on $p = 96$ sampling classes. From this we see the Siamese triplet model is very sensitive to the architecture used. Comparing our implementations (Table 3.3, rows 2 and 3) to their implementation (Table 3.3, row 1), we see that our implementation of their architecture (Table 3.3, row 3) performs almost the same as their implementation (Table 3.3, row 1). However, our architecture (Table 3.3, row 2) significantly underperforms in comparison to theirs. Although our architecture performs worse, we keep this architecture since it is comparable to the architectures of the vision models we use throughout this thesis.

Except for the different Siamese architectures, the remaining difference between our results and that of Eloff et al. [1] in Table 3.2 and Table 3.3, can be attributed to the episode setup. Before training any models, we sample 400 of the following fixed episodes:

- validation episodes from the background data,
- unimodal test episodes from the in-domain data and from the background data,
- multimodal test episodes from the in-domain data.

These episodes are kept fixed across all models, whereas Eloff et al. [1] samples a random episode upon demand. We found that this causes reproducibility issues: model accuracies change depending on the sampled episode instances at that instance. However, our setup

Table 3.3: Our implementation of the Siamese vision models with our architecture vs. Eloff et al.'s [1] architecture trained with $p = 96$.

	Architecture	10-way accuracy (%)	
		one-shot	five-shot
Their implementation	Siamese after mistake fix (from Table 3.1, row 4)	69.46 ± 1.31	87.75 ± 0.69
Our implementation	Our architecture	63.28 ± 1.30	83.98 ± 0.44
	Eloff et al.'s [1] architecture	**68.70 ± 3.21**	**86.29 ± 1.88**

Stellenbosch University https://scholar.sun.ac.za

Table 3.4: Our vision Siamese triplet networks trained with different amounts of sampling classes p.

Amount of sampling classes p	10-way accuracy (%)	
	one-shot	five-shot
96 (from row 1 of Table 3.3)	63.28 ± 1.30	83.98 ± 0.44
88	**64.78** ± **1.60**	**84.75** ± **1.32**

produces more reliable and reproducible results since all models are tested on the same fixed episodes.

3.6.1.2 The Number of Sampling Classes p

For the speech Siamese triplet network, we could only fit a maximum of 88 sampling classes in the memory of a single GPU. Thus, for the speech and vision networks in the multimodal Siamese model to be compatible, we have to use 88 sampling classes for both the speech and vision Siamese triplet networks. Table 3.4 shows that different values of p for the vision Siamese networks, leads to similar image classification accuracy scores. Therefore, using $p = 88$ does not negatively influence the performance of the vision Siamese triplet network.

3.6.2 K-Shot Unimodal Classification Tasks

Before getting to the multimodal matching results, we firstly evaluate the multimodal models' speech and vision networks separately on unimodal classification tasks. Table 3.5 presents the unimodal one- and five-shot 11-way speech classification results for the speech classifier and Siamese networks against a DTW baseline. It shows that both transfer learning models outperform the DTW baseline with the classifier RNN achieving the highest accuracies on the one- and five-shot speech classification tasks.

Table 3.6 shows the results for the pixel baseline and the vision classifier and Siamese networks on unimodal one- and five-shot 10-way image classification tasks. From these

Table 3.5: Unimodal one- and five-shot speech classification accuracies of the unimodal speech transfer learning models vs. a DTW baseline.

Model		11-way accuracy (%)	
		one-shot	five-shot
Baseline	DTW	65.90 ± N/A	89.45 ± N/A
Transfer learning models	Classifier RNN	**86.87** ± **0.83**	**95.40** ± **0.50**
	Siamese RNN	83.52 ± 2.56	94.34 ± 0.86

Stellenbosch University https://scholar.sun.ac.za

Table 3.6: Unimodal image transfer learning models vs. a pixel baseline on unimodal one- and five-shot image classification tasks.

Model		10-way accuracy (%)	
		one-shot	five-shot
Baseline	Pixels	44.58 ± N/A	67.75 ± N/A
Transfer learning models	Classifier CNN	63.23 ± 1.42	82.90 ± 1.12
	Siamese CNN	**64.78 ± 1.60**	**84.75 ± 1.32**

results, we see the trend seen in Table 3.5 does not hold for our vision networks: both transfer learning models outperform the pixel baseline, but the Siamese CNN achieved the highest accuracies on the one- and five-shot image classification tasks.

Since the trends in the unimodal speech and image classification accuracies differ, we conclude that the best unimodal one- or few-shot architecture (classifier or Siamese) is dependent on the modality it is applied to. However, a recent study by Tian et al. [38] found that a simple classifier-like model performed best on a five-way one- or five-shot image classification task. Therefore, we note the classifier might be the best approach for unimodal few-shot classification until the data classes becomes harder to differentiate between. The relative distance learned by the Siamese triplet network might just work better for the image digit classes since the digits are harder to differentiate between, e.g. a *3* and an *8* looks very similar.

3.6.3 K-Shot Multimodal Matching

Finally, we get to the results of the main task considered in this thesis: the multimodal speech-to-image matching results. The models presented here is a re-implementation of the models developed by Eloff et al. [1] in a consistent framework, and will serve as the baselines in all subsequent chapters. After glueing the speech and vision networks together to form the multimodal few-shot models, we use these models to perform the one- and five-shot 11-way speech-to-image matching tasks with the results shown in Table 3.7.

Table 3.7: Multimodal transfer learning models on multimodal one- and five-shot speech-to-image matching tasks.

Model		11-way accuracy (%)	
		one-shot	five-shot
Baseline	DTW + Pixels	31.80 ± N/A	41.88 ± N/A
Transfer learning models	Classifier	**56.80 ± 1.19**	**59.67 ± 1.73**
	Siamese	54.83 ± 1.80	59.25 ± 0.79

Stellenbosch University https://scholar.sun.ac.za

Both transfer learning models outperform the DTW and pixel baseline on the one- and five-shot multimodal matching tasks. This proves that the use of DL is in fact beneficial for this multimodal matching task. As we can see from Table 3.7, despite the fact that the classifier only slightly outperforms the Siamese model, the classifier proved to be the most accurate model on the one- and five-shot matching tasks.

Comparing Table 3.7, Table 3.5 and Table 3.6 we see that the speech-to-image matching scores are consistently lower than the unimodal (speech and image) classification scores. This leads us to conclude that the there is a compounding of errors across the multimodal support set.

Further speech-to-image matching analysis on the appropriate models of this chapter, is done in the next chapter. This chapter simply intends to show that we were able to construct a robust experimental framework and that our results are reliable.

3.7 Chapter Summary

This chapter introduced the multimodal speech-to-image matching task and how a previous study used transfer learned multimodal few-shot learning models to do this task. We re-implemented the experiments in [1] and found that Eloff et al.'s [1] results are not reproducible. After investigation we found that this was due to episodes being randomly sampled, as well as a mistake in the training setup (the same data was used for training and validation). We based our experimental setup from Eloff et al.'s [1], but improved this setup by sampling fixed episodes before any training is done. This leads to a reproducible setup since all the models are tested on the same fixed episodes.

Based on our re-implementation, the multimodal classifier was identified as the best multimodal few-shot model. In Chapter 4, we use this multimodal classifier as our new baseline for a comparison of unsupervised vs. additional transfer learning models on the same indirect two-step approach to do the speech-to-image matching task (Chapter 3.1.2) used in this chapter. In Chapter 5, we will compare these models which follows an indirect matching approach to models which learns a single multimodal embedding space to do the matching task in a direct manner.

Stellenbosch University https://scholar.sun.ac.za

Chapter 4

Unsupervised vs. Transfer Learning for Multimodal Few-Shot Learning

In this chapter we consider unsupervised models to learn representations for the multimodal speech-to-image matching task introduced in Chapter 3.1. This can be motivated by the theory that, before a child is shown new examples of visual objects paired with corresponding spoken words to do a speech-to-image matching task, the child could be exposed to a large amount of unlabelled speech and visual data from its environment. Some of these unlabelled examples could correspond to the classes of the example pairs in the speech-to-image matching task. Motivated by this observation, we ask how unsupervised models trained on unlabelled in-domain data compares to transfer learning from background data (the approach followed in Chapter 3 and by previous work [1]) to do the speech-to-image matching task.

Unsupervised learning is a more difficult learning task than supervised learning since unsupervised models are trained without labelled data. However, this gives the model more flexibility to find the natural structure emerging from the data itself [63]. The hope is that this natural structure can be used to find similar representations for instances of the same unlabelled few-shot class. For our specific case of multimodal few-shot matching, these unsupervised models have the benefit that they are trained on unlabelled in-domain data. This means that, although the data is unlabelled, the model sees some examples of the few-shot digit classes from which it might learn in-domain class specific information. In contrast, although transfer learning models can be trained in a supervised way on labelled data, they are trained on background data and therefore never observe the few-shot classes. This raises the question: although unsupervised learning relies on proxy losses, can it perform better than transfer learning for multimodal few-shot matching?

We specifically consider two unsupervised learning strategies, the AE and the CAE trained on pairs discovered in an unsupervised fashion. In Chapter 4.4, these unsupervised

Stellenbosch University https://scholar.sun.ac.za

models are compared to a transfer learned CAE (trained on ground truth background pairs) as well as the best transfer learned model from Chapter 3, the classifier.

The unsupervised and additional transfer learned multimodal few-shot learning models are combined in the same manner as the models in Chapter 3, e.g. an unsupervised multimodal CAE consists of an unsupervised speech CAE and an unsupervised vision CAE. After introducing the unsupervised multimodal models in Chapter 4.2.1, we discuss the additional transfer learned multimodal models in Chapter 4.2.2. The content of this chapter is published at Interspeech 2020 in a paper entitled *Unsupervised vs. transfer learning for multimodal one-shot matching of speech and images* [64].

4.1 Related Work

For the unimodal speech models in this chapter, we consider unsupervised or transfer learned sequence-to-sequence RNN AEs and CAEs. These models are used to produce a fixed dimensional representation embedding for a given variable duration spoken word consisting of a sequence of acoustic frames. The motivation to find fixed dimensional representations for variable length word instances is two fold: (1) it enables us to compare the variable length word instances that occurs in a speech-to-image matching episode, to each other, and (2) we can remove nuisance information like speaker information and channel noise from the representations to find similar representations for same class word instances. Essentially, this is also the goals of acoustic word embedding models [18, 65–67]. To accomplish this, all these acoustic word embedding studies use RNNs to map words consisting of variable length acoustic frame sequences, to fixed-size embeddings.

In order for these embeddings to be representative of the word class, Kamper [18] and Chung et al. [65, 68] considered unsupervised methods. Kamper [18] and Chung et al. [65] used sequence-to-sequence AE models with an encoder RNN to encode a variable length word instance to a fixed representation. Thereafter a decoder RNN attempts to reconstruct the input sequence from this representation. The intuition behind this is that the representation would summarise the input word (its class) in such a way that the input can be reconstructed from it. Specifically, Chung et al. [65] used a denoising AE to produce word representations for a query-by-example spoken term detection task. Similarly, Kamper [18] trained an AE as well as a variational autoencoder (VAE) and a CAE (Chapter 2.3.7) on words isolated from entire spoken utterances using unsupervised term discovery. To train the CAE, training pairs are mined in an unsupervised fashion.

This CAE produced more similar acoustic word embeddings for a specific word class than either the AE and VAE. Furthermore, the CAE's embeddings where more successful in identifying word instances of the same class than a DTW baseline (Chapter 3). Holzenberger et al. [67] reaffirmed this by using an AE RNN to learn acoustic word embeddings which outperformed a DTW baseline in a word discrimination task. This strengthened the

Stellenbosch University https://scholar.sun.ac.za

motivation to use speech models which is similar to the acoustic word embedding models.

Similarly to the unimodal speech models, we consider unsupervised or transfer learned AEs and CAEs for the unimodal vision models. For the image instances in a speech-to-image matching episode, instead of RNNs we use CNNs to get fixed smaller dimensional representation embeddings. Koutník et al. [69] and Hinton [70] aimed to reduce the dimensionality of input images by finding fixed dimensional features for images using unsupervised CNN networks. Specifically, Koutník et al. [69] used a CNN encoder to feed smaller dimensional representations to a classifier. Hinton [70] used an AE pretrained as a restricted Boltzmann machine to improve the smaller dimensional representations produced by the standard AE.

Such unsupervised vision studies which uses autoencoder-like architectures to find representations for images that is both representative of the image and reduces the dimensionality of images, are limited. Other unsupervised vision studies [71, 72] used autoencoders to find representations for inpainting (image restoration), which is different to the type of representation we aim to learn. Pathak et al. [71] specifically uses an AE CNN that takes in an image with a cut out patch as its input, and aims to produce the patch at its output. To predict the missing parts in a patch, the smaller dimensional representations should capture the surrounding context of the patch. Similarly, Xie et al. [72] uses a stacked sparse denoising AE to remove noise pixels form an image by using the context information (captured in its representations) around the pixel.

4.2 Unsupervised and Transfer Learning Models

Similarly to Chapter 3, in this chapter we also use an indirect approach to do multimodal few-shot learning: as described in Chapter 3.1.2, we use two unimodal comparisons (a speech-speech and a image-image comparison) and a multimodal support set to perform the multimodal few-shot matching task. In this section we therefore discuss different methods to learn features in a single modality. Specifically, we consider unimodal models in two settings: unsupervised models trained on unlabelled in-domain data (Chapter 4.2.1) and transfer learned models trained on labelled background data (Chapter 4.2.2). We compare the performance of these models on a multimodal few-shot matching task to the baselines established in Chapter 3.

To quickly summarise the intuition behind transfer learning (from Chapter 3), the labelled background speech and image data does not contain any of the few-shot classes seen during test time. Therefore, the transfer learned models should use the knowledge gained from these background classes to generalise to the unlabelled unseen few-shot classes. In contrast, the unlabelled in-domain speech and image data includes unlabelled instances of the few-shot classes we see during test time. These in-domain training instances do not occur exactly in the few-shot test episodes. The unsupervised models should learn

Stellenbosch University https://scholar.sun.ac.za

how to generalise to the unlabelled few-shot classes by using the natural structure of each few-shot class that emerges from the in-domain data.

Specifically for the unsupervised unimodal models, we consider two objective functions, an AE and a CAE. The CAE is generally trained on neighbour pairs obtained from data labels. However, since we do not have labels for the in-domain data, we mine within-modality (speech-speech and image-image) pairs from the unlabelled in-domain data to train unsupervised speech and vision CAEs. This pair mining process will be described in Chapter 4.3.1.

Thereafter, to get a clear comparison of unsupervised learning vs. transfer learning, we train transfer learned variants of the unsupervised autoencoder-like models (on ground truth pairs from the background data). An AE is unsupervised in nature since it does not use labels during training. Therefore, we do not consider a transfer learned AE. We also do not consider unsupervised variants of the classifier and Siamese models of Chapter 3. This choice is based on Kamper et al.'s [62] work which showed that an unsupervised speech CAE outperformed unsupervised classifier and Siamese speech models.

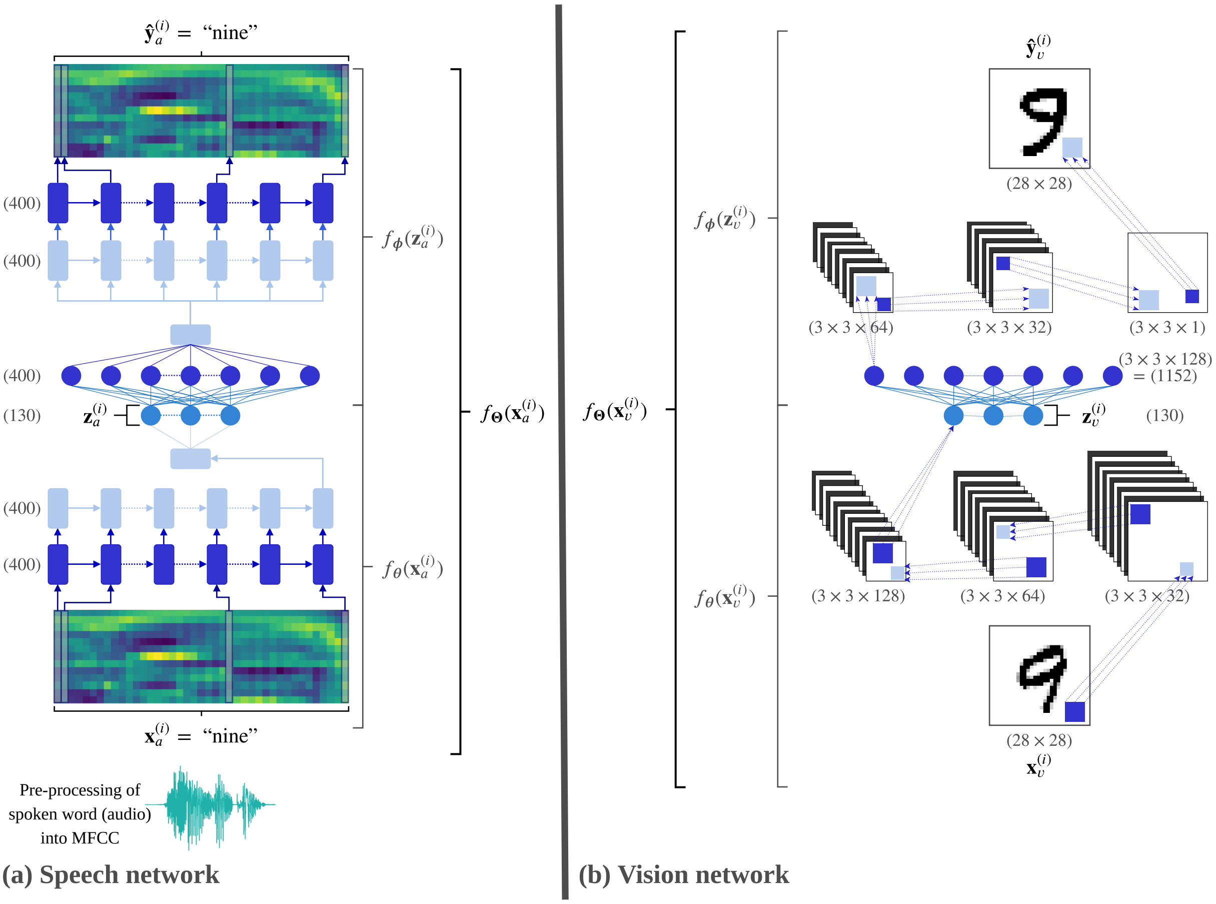

$\hat{\mathbf{y}}_a^{(i)}$ = "nine"
(400)
(400)
(400)
(130)
$\mathbf{z}_a^{(i)}$
(400)
(400)
$f_\phi(\mathbf{z}_a^{(i)})$
$f_\Theta(\mathbf{x}_a^{(i)})$
$f_\theta(\mathbf{x}_a^{(i)})$
$\mathbf{x}_a^{(i)}$ = "nine"
Pre-processing of spoken word (audio) into MFCC
(a) Speech network
$f_\Theta(\mathbf{x}_v^{(i)})$
$f_\phi(\mathbf{z}_v^{(i)})$
$f_\theta(\mathbf{x}_v^{(i)})$
$\hat{\mathbf{y}}_v^{(i)}$
(28 × 28)
(3 × 3 × 64)
(3 × 3 × 32)
(3 × 3 × 1)
(3 × 3 × 128) = (1152)
$\mathbf{z}_v^{(i)}$
(130)
(3 × 3 × 128)
(3 × 3 × 64)
(3 × 3 × 32)
(28 × 28)
$\mathbf{x}_v^{(i)}$
(b) Vision network

Figure 4.1: The AE, CAE and AE-CAE model architectures. (a) A speech RNN is used to learn feature representations for speech data and (b) a vision CNN is used to learn feature representations for image data.

Stellenbosch University https://scholar.sun.ac.za

4.2.1 Unsupervised Models

We consider an unsupervised AE (Chapter 4.2.1.1) and two unsupervised variants of the CAE: a standard CAE (Chapter 4.2.1.2) and an AE-CAE (Chapter 4.2.1.3). The only difference between these three models, as discussed in Chapters 4.2.1.1 to 4.2.1.3, is the specified output $\mathbf{y}$ each network aims to produce. Figure 4.1 shows the architecture for each of the unsupervised (a) speech and (b) vision AE, CAE and AE-CAE networks.

Specifically for the vision networks, we use unlabelled in-domain images to train unsupervised vision networks with the AE, CAE and AE-CAE loss functions. Similarly for the speech networks, we use unlabelled in-domain spoken words to train unsupervised speech networks using the AE, CAE and AE-CAE loss functions.

4.2.1.1 Unsupervised Autoencoder

As a recap from Chapter 2.3.6, a unimodal autoencoder aims to reconstruct its input through a bottleneck feature representation. The multimodal AE consists of a unimodal speech AE as illustrated in Figure 4.1(a) and a unimodal vision AE as illustrated in Figure 4.1(b). For the vision AE, a CNN encoder $f_{\boldsymbol{\theta}}(\mathbf{x}_v^{(i)})$ encodes the input $\mathbf{x}_v^{(i)}$ to the latent representation vector $\mathbf{z}_v^{(i)}$. A decoder with transposed convolutions $f_{\phi}(\mathbf{x}_v^{(i)})$ then decodes $\mathbf{z}_v^{(i)}$ to the network output $\hat{\mathbf{y}}_v^{(i)}$. Similarly for the speech AE, an RNN encoder $f_{\boldsymbol{\theta}}(\mathbf{x}_a^{(i)})$ produces the fixed-sized latent representation vector $\mathbf{z}_a^{(i)}$ which is then used to condition an RNN decoder $f_{\phi}(\mathbf{x}_a^{(i)})$ to produce the network output $\hat{\mathbf{y}}_a^{(i)}$.

Both the speech AE $f_{\boldsymbol{\Theta}}(\mathbf{x}_a^{(i)})$ and the vision AE $f_{\boldsymbol{\Theta}}(\mathbf{x}_v^{(i)})$ are trained with the AE loss function given in Equation 2.9. From this loss function, we see the intuition behind these AEs is that it will produce feature representations $\mathbf{z}^{(i)}$ with only the necessary information (the class) to reconstruct the input.

4.2.1.2 Unsupervised Correspondence Autoencoder

Intuitively the features produced by the above AE will also have to capture unique specifics of the current input besides its class to reconstruct it, e.g. the angle and style of the object in an image input. However, we do not want the feature representations to contain this nuisance information. To overcome this we look at a more complex, and perhaps a more difficult objective to learn: a CAE.

As explained in Chapter 2.3.7, the CAE and AE have identical structures, but instead of attempting to reproduce the input at its output like the AE, the CAE aims to produce a pair of the input at its output through the smaller dimensional (bottleneck) feature representation. The intuition is that the CAE will produce features $\mathbf{z}^{(i)}$ that are invariant to properties not common to the input and the input pair, while only capturing aspects that are (such as the class).

Stellenbosch University https://scholar.sun.ac.za

The multimodal CAE consists of a speech CAE shown in Figure 4.1(a) and a vision CAE shown in Figure 4.1(b). Each unimodal CAE which has an identical encoder-decoder structure as the above unimodal AEs, is trained with the CAE loss function in Equation 2.10. For training, a unimodal CAE requires within-modality input-output pairs where the input instance and output instance of each pair are of the same class and modality. Since our in-domain data is unlabelled, we mine speech-speech and image-image pairs in some unsupervised manner to train the unsupervised speech and vision CAEs.

For the image-image pairs, we mine unsupervised pairs that are predicted to be of the same class by using cosine distance over flattened images from the unlabelled in-domain data. Similarly, DTW over the unlabelled in-domain spoken words are used to find unsupervised speech-speech pairs that are predicted to be of the same class. The image-image and speech-speech pair mining process is discussed in further detail in Chapter 4.3.1.

4.2.1.3 Unsupervised AE-CAE

Lastly we consider the AE-CAE. This model is pretrained with the AE loss function (Equation 2.9) before switching to the CAE loss function (Equation 2.10). More specifically, both the speech AE-CAE and vision AE-CAE are pretrained with the AE loss function before switching to the CAE loss function. The multimodal AE-CAE consists of a speech AE-CAE as shown in Figure 4.1(a) and a vision AE-CAE as shown in Figure 4.1(b). Each unimodal AE-CAE has the same encoder-decoder structure as the above unimodal AEs and CAEs.

The pretraining of the speech and vision AE-CAEs as AEs, are done exactly the same as the training of the unsupervised speech and vision AEs described above. The training of these unimodal AE-CAEs as CAEs, use the same unsupervised mined within-modality pairs than the unimodal unsupervised speech and vision CAEs described above. By pretraining the CAEs as AEs, the hope is that the model will take advantage of the initialisation provided by the AEs to find a better local minimum for the CAE-loss.

4.2.2 Additional Transfer Learning Models

In Chapter 3 we considered transfer learning models using classifiers and Siamese triplet networks. Here we also consider transfer learned variants of the CAE and AE-CAE approaches discussed in Chapter 4.2.1.2 and Chapter 4.2.1.3, i.e. supervised CAE and AE-CAE networks. The difference is that instead of mining unsupervised input-output training pairs, we train these supervised models on ground truth pairs from the background data by using the actual data labels. These transfer learned models were not considered in [1].

The multimodal transfer learned CAE consists of a speech CAE as shown in Figure 4.1(a)

Stellenbosch University https://scholar.sun.ac.za

and a vision CAE as shown in Figure 4.1(b). The unimodal CAEs have an identical encoder-decoder structure as the unimodal CAEs in Chapter 4.2.1.2. However, the unimodal transfer learned CAEs are trained on ground truth pairs from the labelled background data that does not contain any of the few-shot testing classes. For the speech CAE, we use the true word labels of the background data to find input-output pairs so that the input instance and output instance within each pair are from the same word class. Similarly for the vision CAE, we use the true labels of the background training images to find an input instance and an output instance from the same class to form input-output training pairs.

Similarly to the unsupervised multimodal AE-CAE in Chapter 4.2.1.3, the multimodal transfer learned AE-CAE consists of a speech AE-CAE as shown in Figure 4.1(a) and a vision AE-CAE as shown in Figure 4.1(b). Both the unimodal speech and vision AE-CAEs are pretrained with the AE loss function on within-modality (labelled) background data not containing any few-shot classes seen at test time. Although the background data is labelled, the AEs do not require any labels during training. After pretraining the unimodal AE-CAEs, we train these models with the CAE loss function on ground truth within-modality pairs from the background labelled data. The ground truth image pairs are the same pairs used to train the transfer learned vision CAE above. In addition, the ground truth word pairs are the same pairs as the training pairs that the above transfer learned speech CAE is trained on.

4.3 Experimental Setup

Before training the unsupervised CAEs and AE-CAEs, we mine speech-speech pairs for the speech networks and image-image pairs for the vision networks (Chapter 4.3.1). Thereafter we train the unimodal unsupervised and transfer learned models (Chapter 4.2) according to the implementation discussed in Chapter 4.3.2. After training, we pair up corresponding unimodal vision and speech models to construct multimodal few-shot learning models in the same manner as Chapter 3. These models are evaluated on the tasks discussed in Chapter 4.3.3.

4.3.1 Unsupervised Within-Modality Pair Mining

To mine image-image pairs we flatten all the MNIST images from images of size 28×28 pixels to a vector of size 1×784 pixels. We then use cosine distance over the flattened pixels to find images that are most alike. Each image $\mathbf{x}$ in the MNIST dataset is compared to each other image in the same dataset. A given image $\mathbf{x}$ and another image (not $\mathbf{x}$) with the smallest cosine distance to $\mathbf{x}$, are predicted to be of the same class and taken as an image-image pair. We therefore obtained paired training data from the domain in which

Stellenbosch University https://scholar.sun.ac.za

we will be doing few-shot classification, but without access to any labels.

In a similar way, to find speech-speech pairs, we use DTW over the MFCCs of spoken word instances to find word instances that are most similar to one another. Each word instance $\mathbf{x}$ in the TIDigits corpus is compared to every other word instance in the corpus. A word instance $\mathbf{x}$ and another word instance (not $\mathbf{x}$) with the smallest DTW distance score, are predicted to be of the same word class. To ensure that both instances in a speech-speech pair are from different speakers, we take the two word instances from different speakers with the smallest DTW distance as a speech-speech pair. We do this to obtain speaker invariant feature representations for the multimodal few-shot matching task and explicitly consider experiments that measures whether this actually leads to speaker invariant features. The intuition behind this design choice is that it will ensure the unsupervised speech CAE and AE-CAE do not retain any speaker information in their produced feature representations. Logically, if both instances in a speech pair are from different speakers, these speech networks would filter out the speaker information from the representation since it is not something that the input and output pair have in common and therefore would not be helpful to produce the input pair.

4.3.2 Unimodal Model Implementations

All the unsupervised and transfer learned autoencoder-like models have identical architectures for the speech networks shown in Figure 4.1(a) and the vision networks shown in Figure 4.1(b).

Unsupervised speech RNNs are trained using the AE, CAE and AE-CAE loss functions (Chapter 4.2.1) on unsupervised speech-speech pairs which are mined from the unlabelled isolated digit words in the TIDigits training set (Chapter 4.3.1). Unsupervised vision CNNs are trained with the AE, CAE and AE-CAE loss functions (Chapter 4.2.1) on unsupervised image-image pairs mined from the unlabelled digit images in the MNIST training set (Chapter 4.3.1).

The transfer learned speech RNNs are trained using the CAE and AE-CAE loss functions (Chapter 4.2.2) on ground truth speech-speech pairs from the labelled isolated words in the Buckeye training set. The actual Buckeye data labels are used to find speech-speech pairs so that both instances in a pair are from the same word class.

The transfer learned vision CNNs are trained using the CAE and AE-CAE loss functions (Chapter 4.2.2) on ground truth image-image pairs from the labelled character images in the Omniglot training set. Using the Omniglot image labels we setup image-image pairs so that both images in a pair are form the same character class.

We use the same validation and hyperparameter tuning setup discussed in Chapter 3.5.1 to validate and tune the unimodal speech and vision models in this chapter. To obtain the batch size used to train each model, we tune the parameters of each vision model on the

Stellenbosch University https://scholar.sun.ac.za

Table 4.1: The batch sizes used to train each unimodal unsupervised or transfer learned speech or vision network.

Model		Batch size	
		Speech network	Vision network
Unsupervised models	AE	128	1024
	CAE	256	1024
	AE-CAE	256	1024
Transfer learned models	CAE	128	512
	AE-CAE	256	1024

Omniglot test subset and each speech model on the Buckeye test subset. The resulting batch sizes used to train each model, are reported in Table 4.1.

4.3.3 Evaluation Setup

For our main experiments, the multimodal few-shot learning models are tested on the multimodal speech-to-image matching task. For this task, we use the same implementation of the indirect approach to do the speech-to-image matching task given in Chapter 3.5.2.1.

The unimodal speech and vision networks in a multimodal model is also evaluated separately. We do this in order to gain insights into the performance of the different parts of the multimodal models. The unimodal vision networks are evaluated on a unimodal image classification task with this task implementation given in Chapter 3.5.2.3. Chapter 3.5.2.2 describes the unimodal speech classification task implementation that is used to evaluate the unimodal speech networks. In these tasks, we use a confusion matrix to obtain finer analysis of which classes is correctly predicted and which classes the model confuses.

The confusion matrix is employed in either the speech-to-image matching task, the speech classification task or the image classification task. The confusion matrix breaks a model's performance down by reporting for each input given to a model, what is the model's prediction of the input's class and what is the input's actual class. From this matrix we can see a finer breakdown of how exactly classes are misclassified.

From the entries in the confusion matrix we can calculate the recall score. The recall score for a certain class indicates which fraction of all the input queries of this class, is correctly predicted. Specifically, the recall equation for a certain class is:

$$\text{Recall} = \frac{\text{True positives}}{\text{Number of actual positives}}. \tag{4.1}$$

Stellenbosch University https://scholar.sun.ac.za

4.4 Experiments

Firstly in Chapter 4.4.1, as our main experiments, we evaluate the unsupervised and transfer learned multimodal models (Chapter 4.2) on the multimodal speech-to-image matching task. To further investigate what attributes to these results, Chapter 4.4.2 considers the performance of the multimodal models' vision and speech networks in isolation on the unimodal classification tasks. Chapter 4.4.3 goes further by evaluating how good the speech networks are at discarding speaker information. In the last subsections, we then present some experiments towards combining transfer learning and unsupervised learning (details are given at the start of Chapter 4.4.4).

4.4.1 Multimodal K-Shot Speech-to-Image Matching

Our main experiments in Table 4.2 shows multimodal one- and five-shot 11-way results for the multimodal transfer learned and unsupervised models. By comparing the top and bottom sections, we see that on both the one- and five-shot multimodal matching tasks, the transfer learned multimodal classifier outperforms all other unsupervised and transfer learning approaches. None of the unsupervised models perform as well as their transfer learned variants, e.g. the transfer learned CAE has consistently higher one- and five-shot accuracies than the unsupervised CAE. From this we conclude that using prior knowledge from background data results in more useful feature representations for the speech-to-image matching task than using unsupervised learned domain-specific information.

To analyse the results of the multimodal classifier (our best model) further, we consider its confusion matrix on the five-shot 11-way multimodal matching task. By considering the class "one", we see that the classifier mostly predicts queries of the class "one" to have a matching image of a *1*. This holds for all the classes, however, for some classes the model gets more confused than for others: when giving the classifier a query of an

Table 4.2: Multimodal unsupervised vs. transfer learned models on multimodal one- and five-shot 11-way speech-to-image matching tasks.

	Model	11-way accuracy (%)	
		one-shot	five-shot
Transfer learning models	Classifier (from row 2 in Table 3.7)	**56.80 ± 1.19**	**59.67 ± 1.73**
	CAE	46.60 ± 0.69	53.82 ± 1.07
	AE-CAE	48.15 ± 1.21	56.81 ± 1.21
Unsupervised models	AE	28.99 ± 0.84	38.68 ± 1.51
	CAE	42.75 ± 0.62	52.15 ± 0.69
	AE-CAE	42.81 ± 1.01	50.28 ± 0.29

Stellenbosch University https://scholar.sun.ac.za

Table 4.3: The confusion matrix produced by the multimodal classifier on the five-shot 11-way speech-to-image matching task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted image class	*1*	1312	67	50	200	88	87	133	88	127	109	64
	2	36	1053	127	57	13	55	113	78	26	80	87
	3	16	181	1164	14	154	9	99	125	69	62	38
	4	81	39	19	1112	25	119	52	80	139	102	60
	5	32	24	161	27	1143	98	14	152	59	98	44
	6	48	79	28	119	96	1153	9	133	57	152	124
	7	96	162	110	52	23	2	1167	70	196	67	80
	8	73	109	106	80	132	122	60	800	178	116	82
	9	65	40	57	104	43	48	177	174	892	112	93
	0	16	81	23	70	38	102	46	80	57	952	1188
	Total	1775	1835	1845	1835	1755	1795	1870	1780	1800	1850	1860

“eight”, the model often confuses the query as a *9*, *5*, *6* or *3*. Since the word “eight” does not sound acoustically similar to a “nine”, “five”, “six” or “three”, we suspect that the confusion lies in the vision networks since an *8* looks visually very similar to a *9*, *5*, *6* or *3*. To see whether this pattern holds for the other multimodal models, we analyse the results of the classifier, transfer learned CAE and unsupervised CAE on a five-shot 11-way matching task further in Chapter 4.4.1.1.

4.4.1.1 Finer-Grained Analysis

Analysing these results further in order to better understand the differences between the different models, we consider the per-digit recall scores in Table 4.4 for the classifier, transfer learned CAE and unsupervised CAE on the five-shot 11-way multimodal matching task. From these recall scores, we see that for four of the eleven classes, one of the CAE models achieve higher recall scores than the classifier. Therefore, in Table 4.5 we look at sniplets from the confusion matrices of the classifier, transfer learned CAE and unsupervised CAE. Table 4.5 specifically considers the three query classes that the CAEs are the least accurate in predicting correctly: “two”, “five” and “nine”. The entire confusion matrices for the

Table 4.4: The per-digit recall scores of the multimodal classifier vs. multimodal CAEs on a five-shot 11-way speech-to-image matching task.

		Actual speech query digit class										
	Model	“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Recall(%)	Classifier	73.92	**57.38**	**63.09**	**60.60**	**65.13**	**64.23**	**62.41**	44.94	**49.56**	51.46	63.87
	Transfer learned CAE	74.87	45.67	55.18	50.25	41.94	56.77	59.41	43.09	39.50	57.89	**68.82**
	Unsupervised CAE	**78.37**	37.87	51.27	44.14	41.20	47.74	59.20	**45.11**	41.78	**59.78**	65.22

Stellenbosch University https://scholar.sun.ac.za

Table 4.5: Some of the confusion matrix classes produced by the multimodal classifier and CAEs on the five-shot 11-way speech-to-image matching task.

			Predicted image class									
			1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Actual speech digit class	"two"	Classifier	67	1053	181	39	24	79	162	109	40	81
		Transfer learned CAE	158	838	182	53	25	74	202	127	44	132
		Unsupervised CAE	260	695	136	108	27	86	188	112	92	131
	"five"	Classifier	88	13	154	25	1143	96	23	132	43	38
		Transfer learned CAE	146	25	208	73	736	107	71	104	133	152
		Unsupervised CAE	117	17	258	74	723	132	62	127	120	125
	"nine"	Classifier	127	26	69	139	59	57	196	178	892	57
		Transfer learned CAE	159	35	71	240	109	57	159	201	711	58
		Unsupervised CAE	92	68	72	306	140	68	130	126	752	46

transfer learned and unsupervised CAEs can be seen in Table A.1 and Table A.2.

Just considering class "five" in Table 4.5, we see that the classifier mostly predicts a given query of a "five" to belong to the image class of a *3*, *5* or *8*. A "five" and a "three" or "eight" do not sound acoustically similar but they do look very alike. Therefore, we suspect that the confusion between a "five" and an *8* or *3* lies in the vision networks since a *5* and an *8* or *3* are (subjectively) visually similar, but this will be fully discussed in the next section.

The classifier confuses classes less often than the CAEs since it predicts a larger fraction of the queries of a "five" as a *5* (recall 65.13%). From Table 4.5, we see that the transfer learned CAE mostly confuses a query of a "five" as a *3*, *0*, *1* or *9*. This is surprising since neither of these classes are acoustically similar and only a *5* and a *3* or *0* (if the curved part of the *5* is drawn bigger than the rest of the *5*) looks similar. However, it mostly confuses a query of a "five" as a *3*.

Similarly, the unsupervised CAE mostly confuses a query of a "five" as a *3*. Although less, it also often confuses a "five" to be a *6*, *8* or *0*. This is less surprising since although acoustically different, the *5* looks more similar to a *3*, *6*, *8* or *0*. Although the results for the unsupervised CAE are more logical, the transfer learned CAE predicts a slightly larger fraction of the queries of a "five" as a *5*. Such per class trends differ between models.

Furthermore, the models that achieves the highest recall scores for each class, differs. However, the classifier achieves the highest recall scores for most of the classes (Table 4.4). Overall we conclude that the CAEs find less distinctive representations per class for the speech-to-image matching task than the classifier. This is the case irrespective of whether they are trained on out-of-domain labelled background data or in-domain unlabelled data.

Stellenbosch University https://scholar.sun.ac.za

Table 4.6: Unsupervised vs. transfer learning unimodal speech models on unimodal one- and five-shot 11-way speech classification tasks.

Model		11-way accuracy (%)	
		one-shot	five-shot
Transfer learning models	Classifier RNN (from row 2 in Table 3.5)	**86.87 ± 0.83**	**95.40 ± 0.50**
	CAE RNN	79.89 ± 1.32	92.16 ± 0.90
	AE-CAE RNN	80.02 ± 1.04	93.91 ± 0.25
Unsupervised models	AE RNN	53.82 ± 1.70	75.58 ± 1.54
	CAE RNN	75.80 ± 1.76	95.14 ± 0.80
	AE-CAE RNN	77.01 ± 1.29	93.30 ± 0.56

4.4.2 *K*-Shot Unimodal Classification Tasks

In the preceding section we considered the results of the multimodal models by using the indirect approach to do the multimodal matching tasks. In order to obtain finer insights into the performance of these models, we also considered the per-digit performance of some of these models. To further extend this analysis, we now turn to the performance of the individual unimodal models used in the two-step indirect multimodal matching approach. The goal is not to obtain the best possible unimodal results here, but to use the analysis that follows to gain insights into which part of the indirect multimodal matching approach leads to decreased performance.

Table 4.6 shows one- and five-shot 11-way speech classification results. Similar to the trend we see in the multimodal models from Table 4.2, all transfer learning speech models in the top section outperform their unsupervised counterparts in the bottom section. The classifier RNN from Chapter 3 still achieves the highest one- and five-shot speech classification accuracies.

Table 4.7: Unsupervised vs. transfer learning unimodal image models on unimodal one- and five-shot 10-way image classification tasks.

Model		10-way accuracy (%)	
		one-shot	five-shot
Transfer learning models	Classifier CNN (from row 2 in Table 3.6)	**63.23 ± 1.42**	**82.90 ± 1.12**
	CAE CNN	58.23 ± 0.83	78.16 ± 0.87
	AE-CAE CNN	59.36 ± 0.60	79.60 ± 0.60
Unsupervised models	AE CNN	49.71 ± 0.96	66.84 ± 0.99
	CAE CNN	54.98 ± 0.90	77.62 ± 0.69
	AE-CAE CNN	54.41 ± 0.59	76.67 ± 0.91

Stellenbosch University https://scholar.sun.ac.za

Table 4.8: The per-digit recall scores of the speech classifier vs. speech CAEs on a five-shot 11-way speech classification task.

		Actual speech query digit class										
	Model	“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Recall(%)	Classifier	94.02	**99.13**	**97.17**	95.79	**98.25**	**99.95**	**99.24**	**97.37**	98.22	86.13	**97.16**
	Transfer learned CAE	91.36	88.75	96.62	94.37	95.27	97.93	94.85	90.85	92.44	81.55	89.19
	Unsupervised CAE	**95.16**	88.86	96.51	**96.12**	95.38	98.58	98.05	96.66	**98.28**	**88.29**	95.26

The unimodal image classification results seen in Table 4.7 shows a very similar trend to unimodal speech classification (Table 4.6) and multimodal speech-to-image matching (Table 4.2): the transfer learning vision models outperform all the unsupervised vision models with the classifier CNN still achieving the overall highest image classification accuracies.

Considering the per-digit recall scores for the five-shot speech classification tasks as shown in Table 4.8, we see that the scores for the CAE RNNs are quite competitive to the classifier RNN. From the confusion matrices for the three speech models (Table 4.9, Table C.1 and Table C.2) considered in Table 4.8, we see that overall the speech networks produces high recall scores: they less often confuse a query of a certain class to be of another class. From the confusion matrix for the speech classifier RNN on the five-shot 11-way speech classification task in Table 4.9, we see that it rarely confuses classes, e.g. a “six” is misclassified only once as a “seven”. Classes that the speech models sometimes confuses are the classes “nine” and “one”, as well as “two” and “zero”. This makes sense since a “one” and a “nine” ends on the same consonant followed by the same vowel (“ne”). Similarly, a “two” and a “zero” ends on the same vowel.

Table 4.9: The confusion matrix produced by the speech classifier RNN on the five-shot 11-way speech classification task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted speech class	“one”	1730	0	5	0	3	0	0	4	26	58	1
	“two”	0	1824	12	4	0	0	11	1	0	5	31
	“three”	0	2	1754	2	1	0	0	7	0	4	4
	“four”	0	0	10	1753	10	0	0	1	0	58	5
	“five”	2	0	0	12	1744	0	1	3	0	52	0
	“six”	0	0	2	1	0	1834	0	18	0	0	0
	“seven”	1	3	6	3	1	1	1831	0	0	1	8
	“eight”	3	0	13	1	1	0	0	1777	2	40	0
	“nine”	98	0	0	0	5	0	1	9	1768	31	1
	“oh”	5	0	1	21	10	0	0	5	4	1559	1
	“zero”	1	11	2	33	0	0	1	0	0	2	1744
	Total	1840	1840	1805	1830	1775	1835	1845	1825	1800	1810	1795

Stellenbosch University https://scholar.sun.ac.za

Table 4.10: The per-digit recall scores of the vision classifier vs. vision CAEs on a five-shot 10-way image classification task.

		Actual image query digit class									
	Model	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Recall(%)	Classifier	97.35	**77.85**	**85.05**	**82.75**	**86.70**	**84.20**	**83.35**	62.30	**77.95**	91.50
	Transfer learned CAE	98.55	74.80	75.95	71.20	69.85	80.70	80.60	67.30	68.45	**94.25**
	Unsupervised CAE	**98.75**	68.00	79.50	68.05	69.10	75.85	82.70	**70.30**	70.20	93.70

The per-digit recall scores of the CAE CNNs and classifier CNN on a five-shot 10-way image classification task shown in Table 4.10, vary more than the speech scores. Additionally, the image recall scores for the classes *2* to *9* are significanly lower than the scores for the classes *1* and *0* (similarly to the trend seen in the multimodal recall scores). These scores for classes *2* to *9* are also significantly lower than their corresponding speech scores. To investigate why this happens and why this trend is reflected in the multimodal scores, we consider sniplets from the confusion matrices of the vision classifier CNN, transfer learned CAE CNN and unsupervised CAE CNN (Table 4.11) on the classes *2*, *5* and *9* considered in Chapter 4.4.1. The entire confusion matrices for these vision models can be seen in Table E.1, Table E.2 and Table E.3.

Just considering the unsupervised CAE CNN, we see that it mostly confuses an image query of a *5* to be a *3*. This is understandable since these two written characters can look very alike as illustrated in Figure 4.2. However, a "five" and a "three" does not sound acoustically alike. This is reflected by the performance of the unsupervised CAE RNN (Table C.2). The unsupervised CAE RNN only predicted a query of a "five" to be a "three" three times out of the 1775 queries of a "five" that were considered. We conclude that the multimodal unsupervised CAE confuses a query of a "five" to be that of a *3* since its vision network confuses these classes. To prove this for the unsupervised CAE, Figure 4.2

Table 4.11: Some of the confusion matrix classes produced by the vision classifier CNN and CAE CNNs on a five-shot 10-way image classification task.

			Predicted image class									
			1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Actual image digit class	*2*	Classifier	57	1557	93	12	7	28	106	88	4	48
		Transfer learned CAE	106	1496	64	17	12	38	157	66	11	33
		Unsupervised CAE	240	1360	72	51	7	33	141	54	19	23
	5	Classifier	28	0	68	1	1734	74	7	52	10	26
		Transfer learned CAE	63	3	151	26	1397	95	9	91	46	119
		Unsupervised CAE	39	8	192	48	1382	76	13	112	65	65
	9	Classifier	42	12	31	59	19	10	120	116	1559	32
		Transfer learned CAE	34	10	36	208	53	30	103	117	1369	40
		Unsupervised CAE	34	8	30	271	45	27	87	59	1404	35

Stellenbosch University https://scholar.sun.ac.za

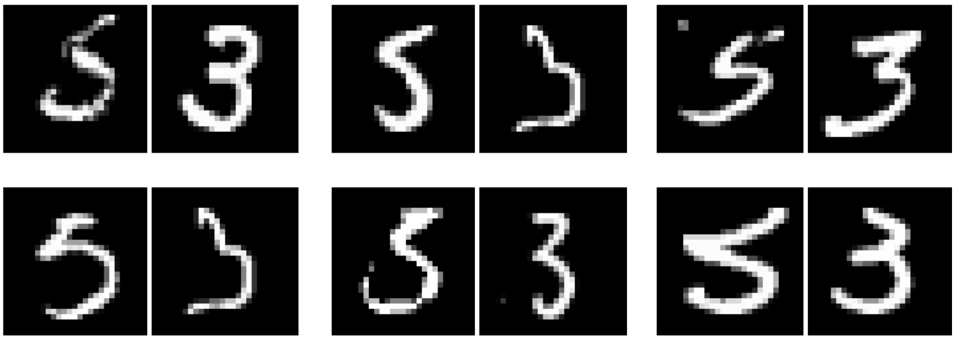


Figure 4.2: Six examples where each pair in the figure shows the multimodal unsupervised CAE predicting a query of a "five" to belong to a support set word instance of a "five", but predicting this instance's paired image of a *5* (left instance in each pair) to be a *3* (right instance of each pair).

shows some of the queries of a "five" which were correctly matched to an instance of a "five" in the support set. These support set instances of a "five" with their paired images of a *5* (left instance in each pair in Figure 4.2) are then confused by the unsupervised vision CNN to belong to images of a *3* (right instance in each pair).

For each of the query classes "two" to "nine", the multimodal models confuse each of these query classes with an incorrect image class because the vision networks confuses these two classes (Table 4.4, Table 4.8 and Table 4.10). This is understandable since the image classes are visually (at a subjective level) more alike than the spoken digit classes.

Comparing the speech classification, image classification and speech-to-image matching accuracies (Table 4.2, Table 4.6 and Table 4.7 or Table 4.4, Table 4.8 and Table 4.10), we see that the multimodal scores are consistently lower than the corresponding unimodal scores. From the analysis we see that the errors made by the vision and speech networks are amplified by their union in the multimodal speech-to-image matching task. I.e. in the two-step indirect matching approach, the speech model could make a mistake in the speech-speech comparisons and pick the wrong (but acoustically similar) item in the support set. This leads to a misclassification even if the image-image comparisons selects the correct image matching the wrong support set item. The speech-speech comparison could also identify the correct support set item, but then the vision model could make a mistake in the image-image comparison by then selecting the wrong image from the test set. Therefore, there is a *compounding* of errors in this two-step approach.

From this discussion, it is evident that the transfer learning approach originally followed in [1] outperforms the unsupervised and transfer learned approaches developed in this chapter.

Stellenbosch University https://scholar.sun.ac.za

4.4.3 Speaker Invariance of the Unimodal Speech Networks

For the indirect multimodal few-shot matching approach, to do the speech-speech comparisons, we need similar representations for spoken word instances. Therefore, a representation for the word "one" said by a speaker should be similar to the representation for the word "one" said by another speaker. A stumbling block in a lot of speech models, is that representations of the word "one" and "done" said by the same speaker would be more similar than the representations of the word "one" said by different speakers. In order to evaluate whether the features produced by our speech models, are invariant to speaker information, we test each unimodal speech network on a harder speech classification task.

In each episode of this speech classification task, we sample only one query and a support set containing K examples for each of the $L = 11$ classes. We sample the instances in the support set in the following manner: all K examples of the same class as the query, are from different speakers than the query. Additionally, the rest of the examples, which are from different classes than the query, are said by the same speaker than the query.

Table 4.12 shows the one- and five-shot 11-way speech classification scores of the speech networks on 400 of these hard speech episodes. From these scores, we see that the classifier RNN achieves the highest accuracies on this hard task. Comparing the top and bottom sections, we also see that the transfer learned speech models outperform their unsupervised variants. We therefore conclude that the classifier produces representations that contains less speaker information than any of the other unsupervised or transfer learned networks. It seems like the classifier is our best model since the word representations of the classifier retains more class information and less speaker information.

Table 4.12: The speaker invariance of the unimodal speech models on hard unimodal one- and five-shot 11-way speech classification tasks.

Model		11-way accuracy (%)	
		one-shot	five-shot
Transfer learning models	Classifier RNN	**84.30 ± 1.19**	**92.45 ± 0.83**
	CAE RNN	62.85 ± 2.68	78.90 ± 0.75
	AE-CAE RNN	59.90 ± 1.34	79.15 ± 1.74
Unsupervised models	AE RNN	30.65 ± 1.67	41.35 ± 1.31
	CAE RNN	55.45 ± 2.82	87.75 ± 1.98
	AE-CAE RNN	50.45 ± 2.47	72.00 ± 3.28

Stellenbosch University https://scholar.sun.ac.za

4.4.4 Towards Combining Transfer and Unsupervised Learning

In the preceding sections we concluded that transfer learning outperforms unsupervised learning on the indirect multimodal matching approach. Despite this conclusion, we ask whether these two methodologies might be complementary: transfer learning might learn certain beneficial general aspects, while unsupervised learning might learn beneficial domain specific aspects. The combination of the two methodologies might lead to better overall performance. Since the direct multimodal matching approach in the next chapter relies on the combination of these two methodologies, we perform an initial investigation by using the combination of these methodologies for the indirect multimodal matching approach.

We propose two models that combine unsupervised and transfer learning: the *Unsupervised CAE with transfer learned classifier pairs* and the *Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs*. Table 4.13 shows the results of these two new combination models we propose. The *Unsupervised CAE with cosine pairs* (Table 4.13, row 2) is repeated from row 5 in Table 4.2.

For the standard unsupervised vision CAE (*Unsupervised CAE CNN with cosine pairs*), we found nearest neighbour image-image pairs using cosine distance (Chapter 4.2.1.2). Instead, to find image-image pairs, we now use cosine distance over the representations for the unlabelled in-domain images where the representations are extracted from the transfer learned vision classifier (trained on background images). We train the *Unsupervised CAE CNN with transfer learned classifier pairs* on these new image-image pairs.

For the *Unsupervised CAE RNN with transfer learned classifier pairs* we find new speech-speech pairs by using cosine distance over the representations for the unlabelled in-domain word instances, where the representations are extracted from the transfer learned speech

Table 4.13: Multimodal one- and five-shot 11-way speech-to-image matching using multimodal models that combine unsupervised and transfer learning.

Model	11-way accuracy (%)	
	one-shot	five-shot
Transfer learning classifier (from row 2 in Table 3.7)	**56.80 ± 1.19**	**59.67 ± 1.73**
Unsupervised CAE with cosine pairs	42.75 ± 0.62	52.15 ± 0.69
Unsupervised CAE with transfer learned classifier pairs	48.66 ± 1.14	55.59 ± 0.71
Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE	54.32 ± 2.19	59.37 ± 1.80
Oracle pairs CAE	89.19 ± 0.69	92.81 ± 0.47

Stellenbosch University https://scholar.sun.ac.za

classifier (trained on background words). Similarly to the standard unsupervised speech CAE (*Unsupervised CAE RNN with cosine pairs*) that used a cosine DTW metric over word instances to find speech-speech pairs (Chapter 4.2.1.2), we use speaker information to ensure that these new speech-speech pairs are from different speakers.

We see that this *Unsupervised CAE with transfer learned classifier pairs* (Table 4.13, row 3) gives a small improvement over the standard CAE (*Unsupervised CAE with cosine pairs*). By additionally initialising speech and vision CAEs by training it on ground truth pairs from the labelled background data and then fine-tuning it on the in-domain classifier generated pairs above (*Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE*, row 4), we get a further improvement.

Although neither of these combination models could outperform the transfer learned classifier (Table 4.13, row 1), performance improved over both the standard unsupervised approach and the transfer learned variants of the unsupervised approach. This trend holds for the confusion matrices (Table A.3 and Table A.4) and recall scores (Table B.1) of these combination models. We also conclude that neither of the combination models could overcome the compounding of errors or find more general representations for the indirect multimodal matching approach than the classifier.

In order to see if it is at all possible to achieve better performance with the CAE by using more accurate training pairs, we also give the accuracy scores (Table 4.13, row 5) of a CAE trained only using correct in-domain pairs. We see that this oracle model outperforms all other approaches, indicating that, if we were able to improve the CAE's training pairs, we might be able to take advantage of an unsupervised learning scheme. Although the oracle model is the most accurate in correctly predicting a query of a certain class, in its confusion matrix in Table A.5 we see there still exists a bit of confusion between certain classes.

In order to see what happens in these combined models, we do a finer-grained analysis in Chapter 4.4.4.1.

4.4.4.1 Understanding the Combined Models

Similar to how we analysed the performance of the transfer learned and unsupervised multimodal models on the indirect multimodal matching approach in Chapter 4.4.2, here we also briefly perform finer-grained analysis of the combined multimodal models on the indirect matching approach. Our goal is to determine whether the trends in the combined transfer learning + unsupervised learning approach are different to those observed for the approaches in isolation.

To gain more insight into the performance of the combined models, we consider the speech and vision networks of the combination models in isolation on the unimodal classification tasks. Table 4.14 shows the one- and five-shot 11-way speech classification scores for the combination models' speech networks. From these results, we see that

Stellenbosch University https://scholar.sun.ac.za

Table 4.14: Unimodal one- and five-shot 11-way speech classification using unimodal speech models that combine transfer and unsupervised learning.

Model	11-way accuracy (%)	
	one-shot	five-shot
Transfer learning classifier RNN (from row 2 in Table 3.5)	86.87 ± 0.83	95.40 ± 0.50
Unsupervised CAE RNN with cosine pairs	75.80 ± 1.76	95.14 ± 0.80
Unsupervised CAE RNN with transfer learned classifier pairs	78.16 ± 2.81	95.75 ± 0.98
Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE RNN	**88.16 ± 1.56**	**97.91 ± 0.30**
Oracle pairs CAE RNN	95.65 ± 0.75	98.67 ± 0.58

differently to the trend we see in the multimodal results (Table 4.13), the *Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE RNN* outperforms the classifier RNN baseline from Chapter 3. From the per-digit speech recall scores of this speech combination model shown in Table D.1, we see that it achieves the highest recall scores for seven of the few-shot classes. The classifier has the highest recall scores for the other four classes. In addition, this combination model achieves recall scores that are very close to oracle results. Since the speech results seems quite promising, we now turn to the vision results to investigate why the combined multimodal models are still outperformed by the multimodal transfer learned classifier.

Table 4.15 shows the one- and five-shot 10-way image classification scores for the vision models. These results follow the same trend as the multimodal results in Table 4.13: the classifier CNN is more accurate than the vision combination models. However, these vision combination models shows improvement over the standard unsupervised CAE CNN

Table 4.15: Unimodal one- and five-shot 10-way image classification using unimodal image models that combine transfer and unsupervised learning.

Model	11-way accuracy (%)	
	one-shot	five-shot
Transfer learned classifier CNN (from row 2 in Table 3.6)	**63.23 ± 1.42**	**82.90 ± 1.12**
Unsupervised CAE CNN with cosine pairs	54.98 ± 0.90	77.62 ± 0.69
Unsupervised CAE CNN with transfer learned classifier pairs	57.57 ± 0.63	79.65 ± 0.69
Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE CNN	60.48 ± 1.63	81.60 ± 1.52
Oracle pairs CAE CNN	94.12 ± 0.49	97.57 ± 0.31

Stellenbosch University https://scholar.sun.ac.za

(Table 4.15, row 2). This is also reflected in the image recall scores in Table F.1. Overall, the image recall scores for some classes (*2* to *9*) are still lower than for others (*1* and *0*), i.e. the vision networks of the combination models does not improve the recall scores of these classes.

The oracle image recall scores in Table F.1 are promising leading us to conclude that improving the image pairs of the unsupervised CAE could lead to better unimodal and multimodal scores. However, recalling the multimodal results of the oracle models above, we notice that there still exists some (although less) confusion when using the very accurate oracle speech and vision networks for the indirect approach to do the speech-to-image task (Table B.1). I.e. the multimodal oracle results have lower recall scores than its vision and speech components because of a compounding of errors in the matching task.

This leads us to ask whether we can get rid of this compounding of errors by reducing the two unimodal comparisons in the indirect speech-to-image matching approach to a single multimodal comparison where speech and image instances can be compared directly in a single embedding space. A direct approach that combines unsupervised and transfer learning to find similar representations for words and images of the same class might also add the necessary information required to more accurately distinguish between the image digit classes.

4.5 Chapter Summary

In this chapter we compared unsupervised and transfer learning models for the multimodal few-shot learning setting. We are the first to consider unsupervised learning for this task. However, the transfer learned models consistently outperformed the unsupervised models on the speech-to-image matching task. After considering the oracle experiments for the unsupervised models, we saw that by improving the unsupervised models' pairs we can find some unsupervised scheme that outperforms the pure transfer learning models. Therefore we combined the unsupervised and transfer learning methodologies by using transfer learning to find pairs for the unsupervised models. We also pretrained one of these unsupervised models on background data before training it on the pairs generated using transfer learning. However, we found these combination models just fell short of the multimodal transfer learned classifier.

On the two-step indirect matching approach, these combination models could not overcome the compounding of errors across the support set that emerged from the purely transfer learned and unsupervised models. In addition, the models considered in this chapter could not find clearly distinguishable features for the image few-shot classes, i.e. the models confuse image classes that are visually too similar.

In an attempt to eliminate this compounding of errors and find better image representations, in the next chapter we consider models that directly maps images and spoken

Stellenbosch University https://scholar.sun.ac.za

words into a single joint multimodal space. Specifically extending the initial experiments in Chapter 4.4.4, we consider combining the unsupervised and transfer learning methodologies to get direct multimodal few-shot learning solutions to do the multimodal speech-to-image matching task using a direct approach.

Stellenbosch University https://scholar.sun.ac.za

Chapter 5

Direct Multimodal Few-Shot Learning

In the previous chapters we followed an indirect approach to do multimodal few-shot learning of speech and images: a speech network measures similarity between spoken words and a vision network measures similarity between images. At test time, we use these networks to do speech-speech and image-image comparisons across a multimodal few-shot support set to indirectly match unseen unlabelled word queries to unseen unlabelled matching images.

> **Indirect Multimodal Few-Shot Learning** involves learning two separate unimodal spaces and using a multimodal few-shot support as a pivot between the two unimodal spaces.

In contrast, in this chapter we consider direct multimodal few-shot learning models which learns a direct mapping between spoken words and images from only the few examples in the multimodal support set. These direct models can measure similarity between the speech and vision domains in a single joint space, so that a single direct comparison can be used in the multimodal speech-to-image matching task to match unseen unlabelled word queries to unseen unlabelled matching images.

> **Direct Multimodal Few-Shot Learning** refers to the task of learning a single multimodal embedding space from a multimodal few-shot support set so that observations from the two modalities can be directly compared.

For instances of the two modalities to be directly comparable, the multimodal embedding space should map cross-modal instances of the same class to similar representations. Specifically for our setting, this multimodal embedding space should find similar representations for spoken word and image digits of the same class.

By attempting to find modality invariant representations per class, we hope to eliminate the phenomenon that emerged from the unimodal models of Chapter 4: some classes

Stellenbosch University https://scholar.sun.ac.za

are often confused to be of another class since some of their instances appear to be very similar. For instance, the unimodal vision models often confused a *5* and a *3* since some instances of *5*'s and *3*'s are visually similar. However, the unimodal speech models could clearly distinguish between a "five" and a "three" since these two words are acoustically different. The intuition is that during training the direct model will notice that although a *5* and a *3* are visually similar, their corresponding word classes sound different and from this realise that the two images are from two different classes. Therefore, if classes in either one of the two modalities are hard to distinguish between, a direct model uses the complementary speech and vision signals to hopefully learn a distance metric that can better distinguish between classes.

This manner of using complementary speech and vision signals to reduce confusion between classes of either a speech or a vision model, is motivated by how humans learn. Borovsky et al. [11] theorised that humans use specific information present in an object to learn its corresponding word, or vice versa [11]. For example, when humans learn the name of a novel dog breed, they might use visual information specific to the breed (e.g. the colour and size of the specific dog breed) to learn the word. Furthermore, children are able to learn a new spoken word from its corresponding visual object, or vice versa, from only a few paired examples [13]. Before seeing these paired examples it is plausible that a child might have seen or heard unlabelled instances of these paired examples. This in-domain specific knowledge obtained in an unsupervised fashion or prior knowledge gained from learning other classes, might aid them in learning these new words and objects.

Therefore, to obtain direct multimodal few-shot learning models, we combine transfer learning with unsupervised learning: unimodal speech and vision transfer learned models, along with a multimodal few-shot support set, are used to mine unsupervised cross-modal (speech-image) pairs from unlabelled in-domain data. On these unsupervised mined speech-image pairs, we train multimodal models which should learn similar representations for cross-modal instances of the same class.

For these multimodal models we consider two multimodal networks: a multimodal correspondence autoencoder (MCAE) discussed in Chapter 5.3.1 and a multimodal triplet network (MTriplet) discussed in Chapter 5.3.2. These multimodal few-shot learning models are used in a direct approach to do the speech-to-image matching task. In Chapter 5.5 we then compare these direct few-shot models to the transfer learned and unsupervised multimodal few-shot models of Chapter 4.

Stellenbosch University https://scholar.sun.ac.za

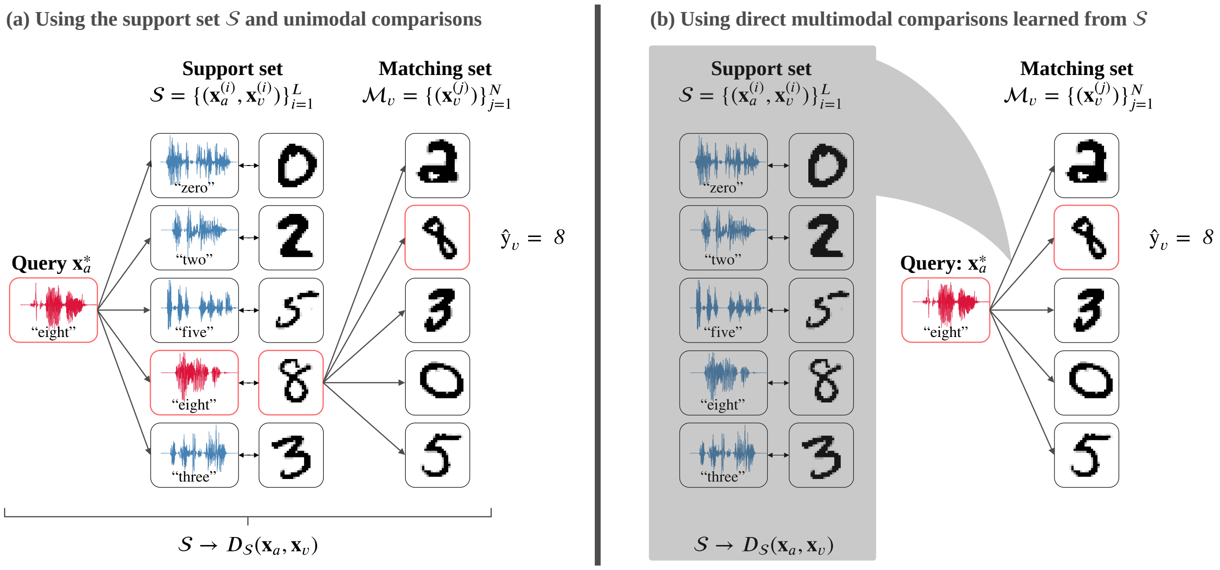

(a) Using the support set $\mathcal{S}$ and unimodal comparisons
Support set
$\mathcal{S} = \{(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})\}_{i=1}^{L}$
Matching set
$\mathcal{M}_v = \{(\mathbf{x}_v^{(j)})\}_{j=1}^{N}$
Query $\mathbf{x}_a^*$
"eight"
"zero"
"two"
"five"
"eight"
"three"
$\hat{y}_v = 8$
$\mathcal{S} \rightarrow D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$
(b) Using direct multimodal comparisons learned from $\mathcal{S}$
Support set
$\mathcal{S} = \{(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})\}_{i=1}^{L}$
Matching set
$\mathcal{M}_v = \{(\mathbf{x}_v^{(j)})\}_{j=1}^{N}$
"zero"
"two"
"five"
"eight"
"three"
Query: $\mathbf{x}_a^*$
"eight"
$\hat{y}_v = 8$
$\mathcal{S} \rightarrow D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$

Figure 5.1: (a) Indirect multimodal one-shot speech-to-image matching using a multimodal support set and two unimodal comparisons (Chapter 3 and Chapter 4), and (b) direct multimodal one-shot speech-to-image matching (Chapter 5).

5.1 A Direct Approach to Multimodal Speech-to-Image Matching

To do the multimodal speech-to-image matching task discussed in Chapter 3.1.1 at test time, a multimodal few-shot learning model $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$ is prompted to match an unseen unlabelled speech query $\mathbf{x}_a^*$ to a matching image from a matching set $\mathcal{M}_v = \{(\mathbf{x}_v^{(j)})\}_{j=1}^{N}$ of unseen unlabelled images. For the direct approach to do this matching task, we find a direct distance metric $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$ between cross-modality (as well as within-modality) inputs to directly match speech queries to matching images. As illustrated in Figure 5.1(b), we find this metric using only a multimodal K-shot support set $\mathcal{S}$ consisting of K isolated spoken words $\mathbf{x}_a^{(i)}$ each paired with a corresponding image of the same class $\mathbf{x}_v^{(i)}$ for each of the L classes. Therefore, we need some direct multimodal few-shot learning model which uses a multimodal support set $\mathcal{S}$ to learn a single joint (multimodal) space from which we obtain a direct distance metric $D_{\mathcal{S}}(\mathbf{x}_a, \mathbf{x}_v)$. I.e. in this multimodal space, cross-modal inputs of the same class should have similar representations. In Chapter 5.3 we discuss the different direct multimodal few-shot models we consider to learn this multimodal space.

Specifically to perform this direct matching approach, from these direct few-shot models we extract representations $\mathbf{z}_a^*$ for each speech query $\mathbf{x}_a^*$ and representations $\mathbf{z}_v^{(v)}$ for each image $\mathbf{x}_v^{(i)}$ in $\mathcal{M}_v$. Thereafter, we compare the representation $\mathbf{z}_a^*$ of each speech query $\mathbf{x}_a^*$ to the representations $\mathbf{z}_v^{(i)}$ of each test image $\mathbf{x}_v^{(i)}$ in $\mathcal{M}_v$. The image $\mathbf{x}_v^{(i)}$ with a representation $\mathbf{z}_v^{(i)}$ that has the smallest cosine distance to the query's representation $\mathbf{z}_a^*$, is chosen as the query's matching image. Figure 5.2 illustrates this multimodal $\mathbf{z}$-space where the query representation of a "three" lies close to the image representation of a *3* and the query

Stellenbosch University https://scholar.sun.ac.za

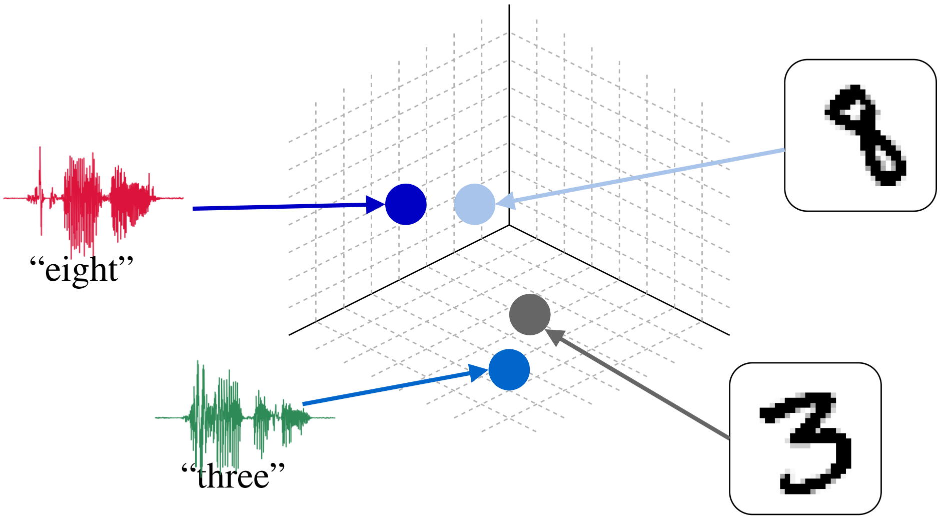


Figure 5.2: The multimodal **z**-space maps spoken words and images of the same class to similar representations.

representation of an "eight" lies close to the image representation of an *8*.

5.2 Related Work

Although we consider multimodal few-shot learning, we need some multimodal network which can jointly model two modalities and their relationship to one another. There is a rich history of multimodal models which are trained to directly represent two different modalities within a single shared space [63, 73–76]. Most importantly for us is the recent work by Harwath et al. [63, 75, 76]. We should emphasise that these studies do not consider multimodal few-shot matching, but (typically) train their models on large amounts of paired data in the two modalities.

Harwath et al. [63, 76] attempts to find a relative similarity metric between images and spoken audio captions by mapping images and their spoken audio captions to joint [63] or separate [76] image and speech representations. This is done by using a multimodal triplet hinge loss and two CNN subnetworks, a speech network and a vision network. The multimodal triplet loss combines two unimodal triplet hinge losses in order to get a relative distance metric between cross-modal inputs: it finds a mapping in which cross-modal observations from the same class are closer to one another than cross-modal observations from different classes. Similarly, we also consider a speech-vision triplet model trained on a modified version of the multimodal triplet loss used by Harwath et al. [63, 76]. We refer to this model as the MTriplet.

The original model from Harwath et al. [63, 76] uses CNNs for the speech and vision subnetworks, whereas our MTriplet consists of a CNN vision network and an RNN speech network. Harwath et al. [63, 76] uses labels to pair up images with corresponding descriptive captions to train their multimodal models. However, we consider this model in the few-shot

Stellenbosch University https://scholar.sun.ac.za

learning setting by training the model on cross-modal pairs mined from a multimodal support set in an unsupervised fashion. Therefore the focus of the MTriplet is to find a relative distance metric for cross-modal inputs. Instead of this relative distance metric, other multimodal modelling studies aim to just find similar representations for same class observations from different modalities.

Some of these multimodal models consists of two autoencoder-like subnetworks where each subnetwork represents one of the two modalities of interest [77–79]. These subnetworks are then connected to one another by a multimodal loss term connecting their representation layers. Ngiam et al. [79] used AEs, as well as a restricted Boltzmann machine and a deep belief network, to get joint audio and video frame representations. Silberer and Lapata [78], Socher et al. [80] and Weston et al. [81] finds joint image and text representations by using stacked bimodal AEs [78] or some probabilistic models [80, 81]. Feng et al. [77] learns separate image and text representations using a vision AE and a text AE connected at their bottleneck representation layers. Each within-modality AE does not only attempt to reconstruct its same-modality input, but also the cross-modal input given to the other modality AE. We consider a similar multimodal network which we refer to as the MCAE. But instead of AEs we use CAEs and we consider the network in the multimodal speech-image few-shot learning setting to find separate spoken word and image representations: the MCAE is trained on cross-modal pairs mined from a multimodal few-shot support set. Furthermore, each CAE in the MCAE does not attempt to produce the other modality CAE's output as well.

As far as we know, we are the first to use the MCAE structure. Additionally, this is only the second study that considers direct multimodal few-shot learning. The first study was a preliminary study of direct multimodal few-shot learning by Eloff [82], which combined the transfer learning and meta-learning approaches to train a model with a similar structure than our MTriplet. Differently to Eloff [82], we train the MTriplet on cross-modal pairs mined in an unsupervised manner. Our direct multimodal study has not yet been compared to Eloff's [82], but this should be done in future work (see the discussion in Chapter 6.4).

5.3 Direct Multimodal Few-Shot Learning

In Chapter 3 and Chapter 4 we considered multimodal models consisting of separate unimodal speech and vision networks. These multimodal models where used to do the multimodal speech-to-image matching task in an indirect two-step approach. Now we consider multimodal models that find a direct distance metric between spoken words and images by learning a single multimodal embedding space. Using this direct distance metric, we can perform the multimodal speech-to-image matching task using a single direct comparison between speech queries and matching images as illustrated in Figure 5.1(b).

Stellenbosch University https://scholar.sun.ac.za

However, to learn the multimodal embedding space, the multimodal models are only provided with a multimodal support set consisting of a few ground truth speech-image pairs.

Since such a small number of speech-image pairs would not be sufficient to train a model capable of successfully learning a mapping between spoken words and images, we mine cross-modal (speech-image) pairs from unlabelled spoken words and images to train these models on. The process we use to obtain these mined cross-modal pairs using transfer learned unimodal speech and vision models and a multimodal support set, is explained in detail in Chapter 5.3.3.1. Since the speech-image training pairs are mined from unlabelled data and is never checked to be correct, the direct models are trained on unsupervised cross-modal pairs. The classes of these unlabelled cross-modal pairs seen by the multimodal models during training, are also seen at test time. However, the instances seen during training do not occur exactly at test time.

By mining the unsupervised speech-image pairs using transfer learned unimodal models, we combine transfer learning and unsupervised learning to obtain multimodal few-shot learning models. It is important to note that these multimodal models are multimodal few-shot learning models since it learns from only the few speech-image pairs given for each of the few-shot classes seen at test time, in contrast to most of the models mentioned in Chapter 5.2.

We consider two direct multimodal few-shot learning models: the MCAE discussed in Chapter 5.3.1 and the MTriplet discussed in Chapter 5.3.2. Both direct models learn separate speech and image representations which can be directly compared. To do this each direct model aims to learn a multimodal space [63, 83, 84] in which speech and image representations of the same class are mapped to similar latent representations. Both these models rely on paired input, which we obtain in an unsupervised way using the mining process. We first describe the two models, and then describe the mining procedure in much more detail in Chapter 5.3.3.

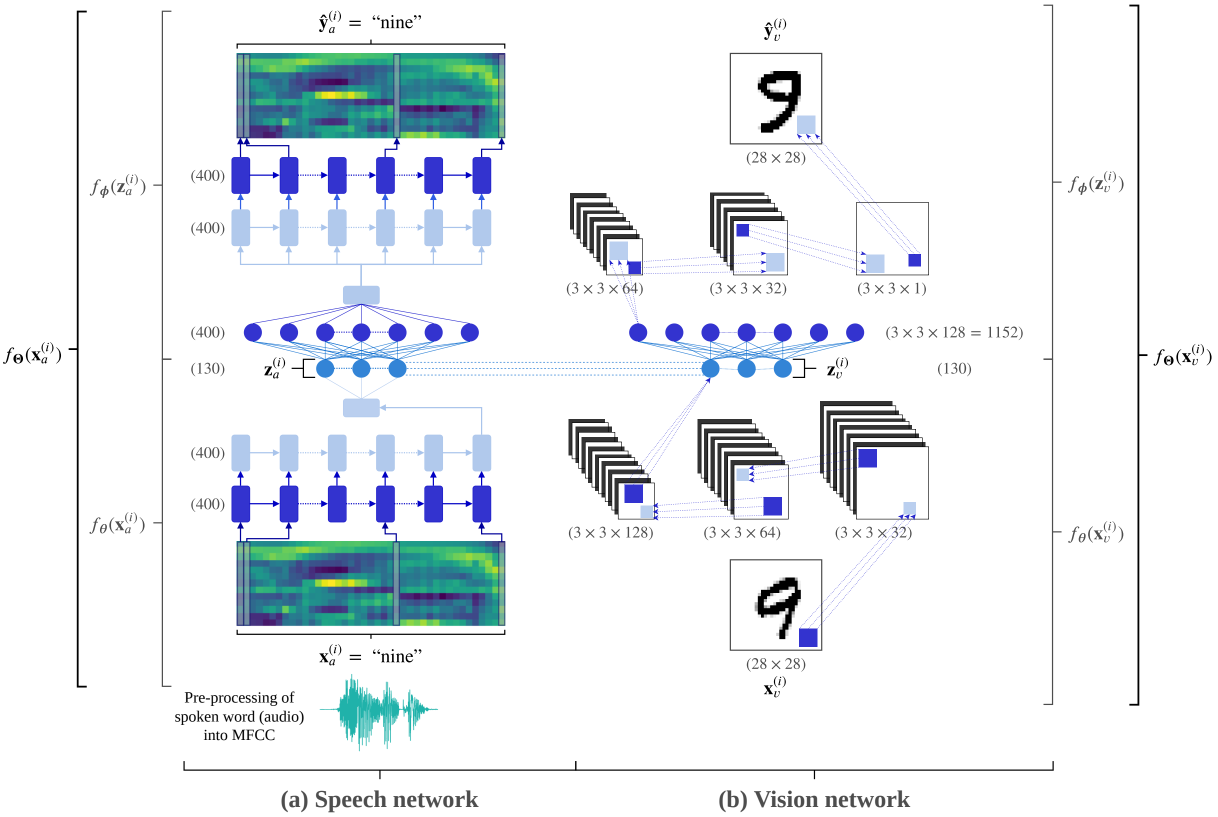
5.3.1 Multimodal Correspondence Autoencoder

The multimodal correspondence autoencoder (MCAE) learns a multimodal embedding space by attempting to learn similar latent representations for speech and image inputs of the same class. To do this the MCAE uses a modified version of the standard CAE loss function (Equation 2.10).

The MCAE consists of two CAEs, a speech CAE RNN $f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)})$ and a vision CAE CNN $f_{\mathbf{\Theta}}(\mathbf{x}_v^{(i)})$ as illustrated in Figure 5.3. Each CAE consists of an encoder which encodes an input to a bottleneck latent representation and a decoder that should ideally decode the latent representation to a pair of the input.

Specifically for the speech subnetwork $f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)})$ of the MCAE, the speech encoder

Stellenbosch University https://scholar.sun.ac.za


$\hat{\mathbf{y}}_a^{(i)}$ = "nine"
$\hat{\mathbf{y}}_v^{(i)}$
(28 × 28)
$f_\phi(\mathbf{z}_a^{(i)})$
$f_\phi(\mathbf{z}_v^{(i)})$
(400)
(400)
(3 × 3 × 64)
(3 × 3 × 32)
(3 × 3 × 1)
(400)
(3 × 3 × 128 = 1152)
$f_\Theta(\mathbf{x}_a^{(i)})$
$f_\Theta(\mathbf{x}_v^{(i)})$
(130)
$\mathbf{z}_a^{(i)}$
$\mathbf{z}_v^{(i)}$
(130)
(400)
(400)
$f_\theta(\mathbf{x}_a^{(i)})$
$f_\theta(\mathbf{x}_v^{(i)})$
(3 × 3 × 128)
(3 × 3 × 64)
(3 × 3 × 32)
$\mathbf{x}_a^{(i)}$ = "nine"
(28 × 28)
$\mathbf{x}_v^{(i)}$
Pre-processing of spoken word (audio) into MFCC

(a) Speech network

(b) Vision network

Figure 5.3: A CNN is used for the vision subnetwork of the MCAE to learn representations for image data and an RNN is used for the speech subnetwork to learn representations for speech data.

RNN $f_{\boldsymbol{\theta}}(\mathbf{x}_a^{(i)})$ encodes the input $\mathbf{x}_a^{(i)}$ to the fixed dimensional latent representation $\mathbf{z}_a^{(i)}$. Thereafter, the speech decoder RNN $f_{\boldsymbol{\phi}}(\mathbf{z}_a^{(i)})$ is conditioned on $\mathbf{z}_a^{(i)}$ to produce the network's output $\hat{\mathbf{y}}_a^{(i)}$. The speech CAE is trained with the standard CAE loss function:

$$\begin{aligned}\ell_a(\mathbf{x}_a^{(i)}, \mathbf{x}_{a_{\text{pair}}}^{(i)}) &= ||\mathbf{x}_{a_{\text{pair}}}^{(i)} - f_{\boldsymbol{\Theta}}(\mathbf{x}_a^{(i)})||_2^2 \\ &= ||\mathbf{x}_{a_{\text{pair}}}^{(i)} - \hat{\mathbf{y}}_a^{(i)}||_2^2.\end{aligned} \tag{5.1}$$

Similarly for the vision subnetwork $f_{\boldsymbol{\Theta}}(\mathbf{x}_v^{(i)})$ of the MCAE, the vision CNN encoder $f_{\boldsymbol{\theta}}(\mathbf{x}_v^{(i)})$ encodes the input $\mathbf{x}_v^{(i)}$ to the latent representation $\mathbf{z}_v^{(i)}$. The vision decoder $f_{\boldsymbol{\phi}}(\mathbf{z}_v^{(i)})$ consisting of transposed convolutions, decodes $\mathbf{z}_v^{(i)}$ to produce the output of the network $\hat{\mathbf{y}}_v^{(i)}$. The vision CNN is also trained with the standard CAE loss function:

$$\begin{aligned}\ell_v(\mathbf{x}_v^{(i)}, \mathbf{x}_{v_{\text{pair}}}^{(i)}) &= ||\mathbf{x}_{v_{\text{pair}}}^{(i)} - f_{\boldsymbol{\Theta}}(\mathbf{x}_v^{(i)})||_2^2 \\ &= ||\mathbf{x}_{v_{\text{pair}}}^{(i)} - \hat{\mathbf{y}}_v^{(i)}||_2^2.\end{aligned} \tag{5.2}$$

Finally, the MCAE is constructed by linking the speech and vision CAEs with a multimodal loss term ℓ_z. The goal of this loss term is to force the speech and image representations for paired speech-image inputs to be similar. For ℓ_z we use a squared loss

Stellenbosch University https://scholar.sun.ac.za

between the speech representation $\mathbf{z}_a^{(i)}$ and image representation $\mathbf{z}_v^{(i)}$:

$$\begin{aligned} \ell_z(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)}) &= ||\mathbf{z}_a^{(i)} - \mathbf{z}_v^{(i)}||_2^2. \\ &= ||f_{\boldsymbol{\theta}}(\mathbf{x}_a^{(i)}) - f_{\boldsymbol{\theta}}(\mathbf{x}_v^{(i)})||_2^2. \end{aligned} \tag{5.3}$$

By combining ℓ_a, ℓ_v and ℓ_z, we obtain the MCAE objective function for a single training example:

$$\ell_{\text{MCAE}}\left(\mathbf{x}_a^{(i)}, \mathbf{x}_{a_{\text{pair}}}^{(i)}, \mathbf{x}_v^{(i)}, \mathbf{x}_{v_{\text{pair}}}^{(i)}\right) = \alpha_a \ell_a(\mathbf{x}_a^{(i)}, \mathbf{x}_{a_{\text{pair}}}^{(i)}) + \alpha_v \ell_v(\mathbf{x}_v^{(i)}, \mathbf{x}_{v_{\text{pair}}}^{(i)}) + \alpha_z \ell_z(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)}), \tag{5.4}$$

where α_a, α_v and α_z are some weighting constants and each MCAE training example consists of $\mathbf{x}_a^{(i)}$, $\mathbf{x}_{a_{\text{pair}}}^{(i)}$, $\mathbf{x}_v^{(i)}$ and $\mathbf{x}_{v_{\text{pair}}}^{(i)}$. In each training example $\mathbf{x}_a^{(i)}$ and $\mathbf{x}_v^{(i)}$ is an unsupervised mined speech-image pair. From this speech-image pair, we mine unsupervised within-modality positive pairs: $\mathbf{x}_{a_{\text{pair}}}^{(i)}$ from $\mathbf{x}_a^{(i)}$ and $\mathbf{x}_{v_{\text{pair}}}^{(i)}$ from $\mathbf{x}_v^{(i)}$.

The intuition is that the reconstruction loss terms will force the model to learn the information common to within-modality inputs of the same class while at the same time the multimodal loss term forces the model to only learn the information common to cross-modal inputs, i.e. the class. This should lead to similar latent representations for speech and image inputs of the same class.

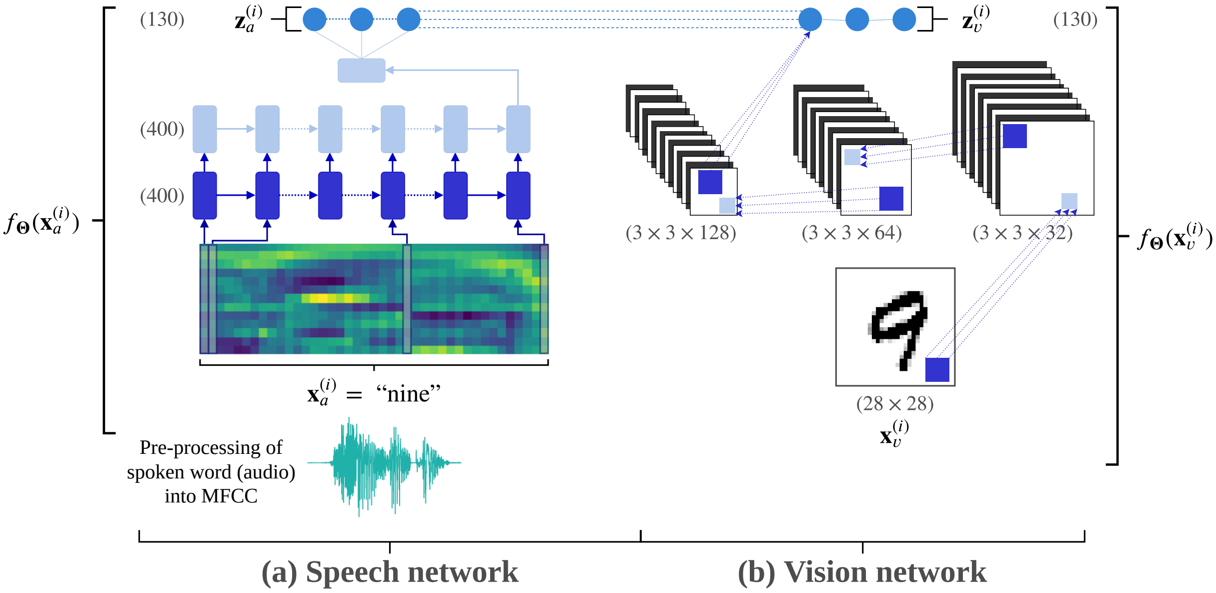


Figure 5.4: A CNN is used for the vision subnetwork of the MTriplet to learn representations for image data and an RNN is used for the speech subnetwork to learn representations for speech data.

Stellenbosch University https://scholar.sun.ac.za

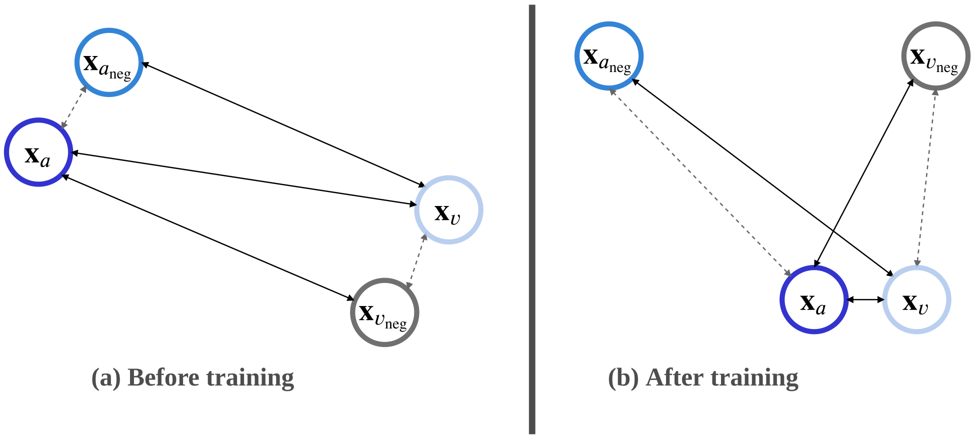
5.3.2 Multimodal Triplet Network

The multimodal triplet network (MTriplet) architecture consists of two subnetworks, a speech RNN network $f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)})$ and a vision CNN network $f_{\mathbf{\Theta}}(\mathbf{x}_v^{(i)})$ as shown in Figure 5.4. The speech subnetwork $f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)})$ encodes a speech input $\mathbf{x}_a^{(i)}$ to a representation $\mathbf{z}_a^{(i)}$. Similarly the vision subnetwork $f_{\mathbf{\Theta}}(\mathbf{x}_v^{(i)})$ encodes an input image $\mathbf{x}_v^{(i)}$ to a representation $\mathbf{z}_v^{(i)}$.

From this we see that the MTriplet draws inspiration from Siamese networks [48, 49]. A Siamese network learns a *relative* distance metric in which the distance between inputs of the same class should ideally be smaller than the distance between inputs from different classes as discussed in Chapter 2.3.5. The MTriplet aims to learn a similarity metric between speech and image inputs by combining two unimodal triplet hinge losses (Chapter 2.3.5) into a multimodal triplet hinge loss [63, 76]. The aim of the multimodal triplet hinge loss is to push cross-modal representations of the same class towards each other while simultaneously pushing cross-modal representations from different classes away from each other as illustrated in Figure 5.5. Specifically, the distance between the representations of inputs from the same class ($\mathbf{x}_a^{(i)}$ and $\mathbf{x}_v^{(i)}$) should to be smaller than the distance between the representations of inputs from different classes ($\mathbf{x}_a^{(i)}$ and $\mathbf{x}_{v_{\text{neg}}}^{(i)}$, as well as $\mathbf{x}_{a_{\text{neg}}}^{(i)}$ and $\mathbf{x}_v^{(i)}$).

We modify the loss used in Harwath et al. [63, 76] to obtain our version of a multimodal triplet hinge loss ℓ_{MTriplet}:

$$\begin{aligned}\ell_{\text{MTriplet}}(\mathbf{x}_a^{(i)}, \mathbf{x}_{a_{\text{neg}}}^{(i)}, \mathbf{x}_v^{(i)}, \mathbf{x}_{v_{\text{neg}}}^{(i)}) &= \max\left\{0, m + d(\mathbf{z}_a^{(i)}, \mathbf{z}_v^{(i)}) - d(\mathbf{z}_a^{(i)}, \mathbf{z}_{v_{\text{neg}}}^{(i)})\right\} \\ &\quad + \max\left\{0, m + d(\mathbf{z}_a^{(i)}, \mathbf{z}_v^{(i)}) - d(\mathbf{z}_{a_{\text{neg}}}^{(i)}, \mathbf{z}_v^{(i)})\right\} \\ &= \max\left\{0, m + d(f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)}), f_{\mathbf{\Theta}}(\mathbf{x}_v^{(i)})) - d(f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)}), f_{\mathbf{\Theta}}(\mathbf{x}_{v_{\text{neg}}}^{(i)}))\right\} \\ &\quad + \max\left\{0, m + d(f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)}), f_{\mathbf{\Theta}}(\mathbf{x}_v^{(i)})) - d(f_{\mathbf{\Theta}}(\mathbf{x}_{a_{\text{neg}}}^{(i)}), f_{\mathbf{\Theta}}(\mathbf{x}_a^{(i)}))\right\}\end{aligned} \tag{5.5}$$


$\mathbf{x}_{a_{\text{neg}}}$
$\mathbf{x}_a$
$\mathbf{x}_v$
$\mathbf{x}_{v_{\text{neg}}}$
$\mathbf{x}_{a_{\text{neg}}}$
$\mathbf{x}_{v_{\text{neg}}}$
$\mathbf{x}_a$
$\mathbf{x}_v$

(a) Before training

(b) After training

Figure 5.5: The logic behind the MTriplet loss function.

Stellenbosch University https://scholar.sun.ac.za

where m is a margin parameter and

$$d(\mathbf{z}_1, \mathbf{z}_2) = 0.5 \times \left(1 - \frac{\mathbf{z}_1 \cdot \mathbf{z}_2}{||\mathbf{z}_1||\,||\mathbf{z}_2||}\right) \tag{5.6}$$

is the cosine distance between representations $\mathbf{z}_1$ and $\mathbf{z}_2$. Therefore, a single MTriplet training example consists of $\mathbf{x}_a^{(i)}$, $\mathbf{x}_{a_\text{neg}}^{(i)}$, $\mathbf{x}_v^{(i)}$ and $\mathbf{x}_{v_\text{neg}}^{(i)}$, where $\mathbf{x}_a^{(i)}$ and $\mathbf{x}_v^{(i)}$ are mined speech-image pairs from the same class and $\mathbf{x}_{a_\text{neg}}^{(i)}$ and $\mathbf{x}_{v_\text{neg}}^{(i)}$ are mined negative pairs from any other class than $\mathbf{x}_a^{(i)}$ and $\mathbf{x}_v^{(i)}$. This means that all the positive and negative pairs within a training example, is obtained in an unsupervised manner.

The intuition behind the MTriplet is that it will learn to distinguish between inputs from the same class and inputs from different classes regardless of which modalities the inputs are from.

5.3.3 Pair Mining

Since we train the MCAE and MTriplet as few-shot learning models, the only ground truth pairs we are provided with, is the speech-image pairs in the given multimodal support set $\mathcal{S}$. This small set of pairs would not be sufficient for training a multimodal model. We therefore use this multimodal support set $\mathcal{S}$ to *mine* speech-image pairs from a larger set of unlabelled in-domain data.

5.3.3.1 Cross-Modal Pair Mining

To obtain speech-image pairs from the unlabelled in-domain data, we use the multimodal support set $\mathcal{S}$ to mine pairs. More concretely, we use the support set $\mathcal{S}$ as a pivot between the unlabelled data in the two modalities. Figure 5.6 illustrates this mining process. For instance, using the support set pair of an eight (the third item in the support set), we find the images in the in-domain image dataset whose closest image in the support set, is the image of the *8*. Similarly for spoken word instances in the in-domain speech dataset, we find the word instances whose closest word instance in the support set, is the word instance of the "eight". From these word and image instances matched to the pair of an eight, we choose a speech instance and an image instance and pair them up. Figure 5.6 shows that some of these pairs are correct like the paired speech-image pair of an eight, while some could be incorrect like the speech-image pair that should consist of an "eight" and an *8* but instead consists of an "eight" and a *3*.

In order to mine speech-image pairs from the multimodal support set $\mathcal{S}$, we need speech-speech and image-image metrics. We use the transfer learning methodology to learn these metrics. More specifically, we use the speech and vision classifiers (trained on background labelled data) from Chapter 3 to extract representations for unlabelled in-domain speech and image inputs. The hope is that these unimodal models produce

Stellenbosch University https://scholar.sun.ac.za

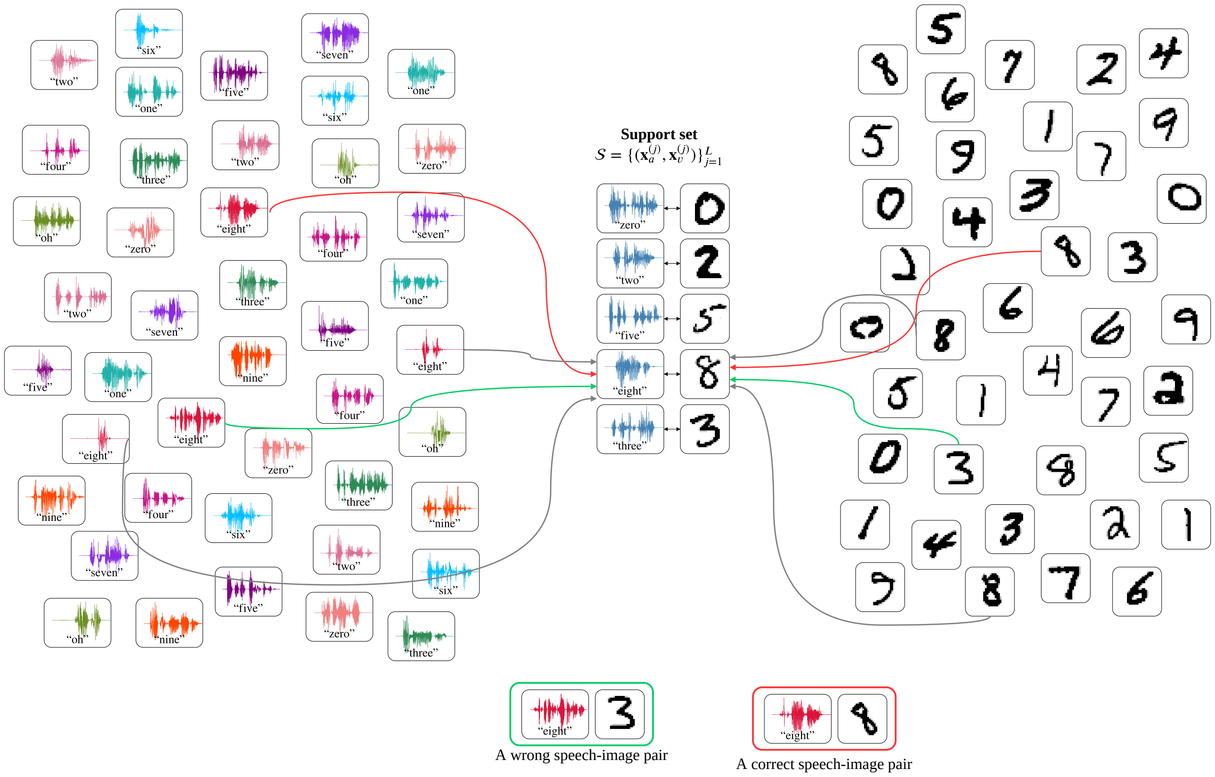


Figure 5.6: After all image and word instances in the datasets are matched to a support set pair, a random image and word matched to a pair in the support set, is paired up. Labels are shown purely for illustrative purposes. Since we use no labels to pair up the spoken word and images, all the pairs would not be correct.

similar representations for within-modality inputs of the same class. Unlabelled spoken words are fed to the speech classifier from which the representation layer $\mathbf{z}$ is extracted to represent the given speech input. Similarly, unlabelled images are fed to the vision classifiers from which the representation layer $\mathbf{z}$ is extracted to represent the given image.

We extract representations for all the spoken words and images in the in-domain speech and image datasets, as well as the word and image instances in the sampled multimodal support set $\mathcal{S}$. The word and image instances in $\mathcal{S}$ does not occur exactly in the in-domain speech and image datasets. To mine speech-image pairs, we use the extracted representations in unimodal speech-speech and image-image comparisons across the sampled few-shot support set $\mathcal{S}$. More specifically, we use a smallest cosine distance metric to match the representation of each speech example in the unlabelled in-domain speech dataset to the representation of a speech instance in $\mathcal{S}$. Similarly, we use a smallest cosine distance metric to match the representation of each image example in the unlabelled in-domain image dataset to the representation of an image instance in $\mathcal{S}$.

To set up the speech-image pairs, we take each speech-image pair in the multimodal support set $\mathcal{S}$ and randomly pick a speech example matched to the pairs' speech instance and an image example matched to the pairs' image instance. The chosen speech example and image example is then used as a speech-image pair $(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})$ as illustrated in Figure 5.6.

Stellenbosch University https://scholar.sun.ac.za

We remove this word and image examples from the matched examples so that they only occur in one speech-image pair. For each pair in the multimodal support set $\mathcal{S}$, this process is repeated untill we run out of speech or image examples matched to the specific support set pair.

As illustrated in Figure 5.6, all the unsupervised mined cross-modal pairs would not be correct since we do not use any labels to check whether these pairs are correct. We expect this will have an effect on the direct models' ability to find similar representations for same class speech and image inputs.

Together with these cross-modal pairs, we also need within-modality (speech-speech and image-image) positive pairs for the MCAE and negative pairs for the MTriplet. This is discussed in the next two subsections.

5.3.3.2 Within-Modality Positive Pair Mining

To train the MCAE in Figure 5.3, we see that besides for the input speech-image pair, we also need an output speech instance and an output image instance from the same class as the input pair. For the multimodal few-shot learning setting we only have access to labels in the given multimodal support set $\mathcal{S}$. Therefore, we cannot use class labels to sample within-modality positive pairs $(\mathbf{x}_a^{(i)}, \mathbf{x}_{a_{\text{pair}}}^{(i)})$ and $(\mathbf{x}_v^{(i)}, \mathbf{x}_{v_{\text{pair}}}^{(i)})$. From a speech-image input pair $(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})$, we mine a positive image pair $\mathbf{x}_{v_{\text{pair}}}^{(i)}$ from its image instance $\mathbf{x}_v^{(i)}$ and a positive word pair $\mathbf{x}_{a_{\text{pair}}}^{(i)}$ from its word instance $\mathbf{x}_a^{(i)}$.

It is important to note that the positive image pairs we mine here are not the same pairs as the image pairs in Chapter 4.4.4 which are mined using the vision classifier. These image pairs from Chapter 4.4.4 did not use the hard restrictions we use in this section. However, the speech positive pairs in this section are mined similarly as the speech pairs in Chapter 4.4.4 which are mined using the speech classifier and hard speaker restrictions.

To mine hard positive speech pairs, we use the transfer learned speech classifier of Chapter 3 (trained on background labelled words) to extract feature representations for all the unlabelled spoken word instances in the in-domain speech dataset. We calculate the cosine distance between the representation of a spoken word $\mathbf{x}_a^{(i)}$ in a speech-image pair $(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})$ and the representation of every other word $\mathbf{x}_a^{(j)}$ in the in-domain speech dataset. Thereafter, we take the spoken word $\mathbf{x}_a^{(j)}$ from a different speaker than $\mathbf{x}_a^{(i)}$ and with a word representation that has the smallest cosine distance to the representation of $\mathbf{x}_a^{(i)}$, as the hard speech pair $\mathbf{x}_{a_{\text{pair}}}^{(i)}$ for $\mathbf{x}_a^{(i)}$.

In a similar manner we find image pairs by first extracting feature representations for all the unlabelled image instances in the in-domain image dataset by using the transfer learned vision classifier (trained on background labelled images) of Chapter 3. We calculate the cosine distance between the representation for the image $\mathbf{x}_v^{(i)}$ in a speech-image pair $(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})$ and the representation of every other image $\mathbf{x}_v^{(j)}$ in the in-domain image dataset. The image $\mathbf{x}_v^{(j)}$ with a representation that has the smallest cosine distance within the range

Stellenbosch University https://scholar.sun.ac.za

of $[0.05, 0.25]$ to the representation of $\mathbf{x}_v^{(i)}$, is taken as the hard image pair $\mathbf{x}_{v_{\text{pair}}}^{(i)}$ for $\mathbf{x}_v^{(i)}$.

The range $[0.05, 0.25]$ is chosen and tuned on the test subset of the developmental image dataset. Logically image pairs with cosine distances in this range should be hard positive image pairs since cosine distance values are in the range of $[0, 1]$, where a value of 0 means the pairs are identical and a value of 1 means the pairs are the most different they could possibly be.

5.3.3.3 Within-Modality Negative Pair Mining

From the MTriplet in Chapter 5.3.2, we see that except for the speech-image input pair, we also need a negative speech instance and a negative image instance from different classes as the input pair.x To sample within-modality negative pairs $(\mathbf{x}_a^{(i)}, \mathbf{x}_{a_{\text{neg}}}^{(i)})$ and $(\mathbf{x}_v^{(i)}, \mathbf{x}_{v_{\text{neg}}}^{(i)})$, we cannot use labels since for multimodal few-shot learning we only have access to labels in the given multimodal support set $\mathcal{S}$. From a speech-image input pair $(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})$, we mine a negative image pair $\mathbf{x}_{v_{\text{neg}}}^{(i)}$ from its image instance $\mathbf{x}_v^{(i)}$ and a negative word pair $\mathbf{x}_{a_{\text{neg}}}^{(i)}$ from its word instance $\mathbf{x}_a^{(i)}$.

To sample hard speech negatives $(\mathbf{x}_a^{(i)}, \mathbf{x}_{a_{\text{neg}}}^{(i)})$, we follow a similar procedure than mining hard speech positive pairs above. We start by using the transfer learned speech classifier (trained on background labelled words) of Chapter 3 to extract feature representations for all the unlabelled spoken word instances in the in-domain speech dataset. Next, we calculate the cosine distance between the representation of a spoken word $\mathbf{x}_a^{(i)}$ in a speech-image pair $(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})$ and the representation of each other word $\mathbf{x}_a^{(j)}$ in the in-domain speech dataset. We take the spoken instance $\mathbf{x}_a^{(j)}$ from the same speaker as $\mathbf{x}_a^{(i)}$ and with a cosine distance in the 50^{th} to 70^{th} percentile of closest cosine distances to the representation of $\mathbf{x}_a^{(i)}$, as the hard negative pair $\mathbf{x}_{a_{\text{neg}}}^{(i)}$ for $\mathbf{x}_a^{(i)}$. These hard percentile constraints are tuned on the test subset of the developmental speech dataset.

To mine hard image negative pairs $(\mathbf{x}_v^{(i)}, \mathbf{x}_{v_{\text{neg}}}^{(i)})$, we use the transfer learned vision classifier (trained on background labelled images) of Chapter 3 to extract feature representations for each unlabelled image example in the in-domain image dataset. Thereafter, the cosine distance between the representation of the image $\mathbf{x}_v^{(i)}$ in a speech-image pair $(\mathbf{x}_a^{(i)}, \mathbf{x}_v^{(i)})$ and the representation of each other image $\mathbf{x}_v^{(j)}$ in the in-domain image dataset, is calculated. The cosine distance between the representation of $\mathbf{x}_v^{(j)}$ and the representation of $\mathbf{x}_v^{(i)}$ which lies in the range of $[0.6, 0.8]$ and in the 50^{th} to 70^{th} percentile of closest cosine distances to the representation of $\mathbf{x}_v^{(i)}$, results in the hard negative image pair $(\mathbf{x}_v^{(i)}, \mathbf{x}_{v_{\text{neg}}}^{(i)})$.

The range $[0.6, 0.8]$ and the hard percentile constraint are tuned on the test subset of the developmental image dataset. Since cosine distance values are in the range of $[0, 1]$, then logically image negative pairs with cosine distances in the range $[0.6, 0.8]$ should be hard negative image pairs.

Stellenbosch University https://scholar.sun.ac.za

5.4 Experimental Setup

In Chapter 5.4.1 we discuss the implementation of the direct multimodal few-shot learning models which is trained on mined speech-image pairs. Thereafter, we evaluate these models using the tasks discussed in Chapter 5.4.2.

5.4.1 Models

Both the MCAE and MTriplet are trained on in-domain mined pairs of isolated spoken digits and handwritten digit images from the MNIST and TIDigits datasets. To mine cross-modal and within-modality training pairs as described in Chapter 5.3.3, we use the MNIST and TIDigits training subsets. Specifically to mine training speech-image pairs, we sample a multimodal five-shot 11-way support set from these training subsets and remove all instances in this support set from the training subsets. Since we use a multimodal five-shot support set to mine speech-image training pairs, the MCAE and MTriplet are multimodal five-shot models.

We validate the MCAE and MTriplet by performing early stopping using the model objective function on in-domain mined validation pairs. For the cross-modal and within-modality validation pairs, we use the MNIST and TIDigits validation subsets. To mine speech-image validation pairs, we specifically sample another multimodal five-shot 11-way support set from these validation subsets and remove all instances in this support set from the validation subsets.

Both the training and validation multimodal five-shot support sets sample five spoken word and image digit pairs for each of the $L = 11$ classes ("one" to "nine", as well as "zero" and "oh"). Chapter 3.5 discusses the training and validation of the transfer learned speech and vision classifiers used for cross-modal and within-modality pair mining.

The MCAE architecture is given in Figure 5.3 with the alphas in the MCAE-loss ℓ_{MCAE} (Equation 5.4) set to: $\alpha_a = 0.3$, $\alpha_v = 0.3$ and $\alpha_z = 0.4$. Figure 5.4 gives the MTriplet architecture and the loss margin m in the MTriplet-loss ℓ_{MTriplet} (Equation 5.5) is set to $m = 0.2$. We do not tune the direct models' hyperparameters, but report model stability over five different batch sizes $\mathcal{B} = \{16, 32, 64, 128, 256\}$.

5.4.2 Evaluation

To evaluate the direct MCAE and MTriplet approaches for multimodal speech-to-image matching, we sample 400 multimodal episodes where each episode samples ten different spoken digit queries and a matching set $\mathcal{M}_v$. As a recap from Chapter 3.5.2.1, there are only ten unique handwritten digit classes. Therefore, we sample ten different digit images for the matching set $\mathcal{M}_v$.

Stellenbosch University https://scholar.sun.ac.za

At test time, each of the ten queries has to be matched directly to the correct image in the matching set as described in Chapter 5.1. Similarly to the indirect approach implementation in Chapter 3.5.2.1, in the direct approach, if a model is given a speech query which is either a "zero" or an "oh", it is counted as correct if the model's matching image prediction is that of a *0*. This is a multimodal five-shot speech-to-image matching task since the model used a five-shot 11-way support set to mine the cross-modal training pairs.

Since we use a multimodal five-shot support set to mine the cross-modal pairs for the direct multimodal few-shot learning models, we have to compare these direct models to the indirect multimodal five-shot learning models. I.e. we use the multimodal five-shot speech-to-image matching accuracies reported in Chapter 3 and Chapter 4. For comparability, we also test the direct models on the exact same five-shot episodes as these indirect models. We just do not use the sampled multimodal five-shot 11-way support set in these episodes.

At test time, all multimodal speech-to-image matching tasks are performed on the TIDigits and MNIST test subsets. The scores reported for the MCAE and MTriplet are averaged over five models each trained with a different batch size. Each model with a specific batch size is also trained with five different seeds. Scores are reported with 95% confidence intervals.

To further investigate the performance of the direct few-shot models, we evaluate the speech and vision subnetworks of a direct model in isolation. The speech subnetworks are evaluated on a unimodal speech classification task using the TIDigits test subset as discussed in Chapter 3.5.2.2. Similarly, we evaluate the vision subnetworks on a unimodal image classification task by using the MNIST test subset as discussed in Chapter 3.5.2.3. Similarly as Chapter 4, we use confusion matrices and per-digit recall scores to aid in further analysis of the results achieved on these unimodal and multimodal tasks.

5.5 Experiments

In Chapter 5.5.1 we evaluate the MCAE and MTriplet on the direct approach to do the speech-to-image matching task. At the same time we compare these models to the multimodal models used to do this task with the indirect approach in Chapter 4.

To obtain further insight, in Chapter 5.5.2 we isolate the speech and vision networks that the direct and indirect few-shot learning models consist of, and test these networks in isolation on unimodal classification tasks. Chapter 5.5.3 considers what effect the unsupervised mined pairs (in which not all the pairs are correct) have on the direct few-shot models. Lastly, Chapter 5.5.4 evaluates whether the representations produced by the direct models' speech networks, are speaker independent.

Stellenbosch University https://scholar.sun.ac.za

Table 5.1: Multimodal five-shot 11-way speech-to-image matching using the direct approach (direct models) vs. the indirect approach (indirect models).

	Model	five-shot 11-way accuracy (%)
Indirect multimodal few-shot learning models	Transfer learned classifier (from Table 4.13 row 1)	59.67 ± 1.73
	Unsupervised CAE (from Table 4.13 row 2)	52.15 ± 0.69
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE (from Table 4.13 row 4)	59.37 ± 1.80
Direct multimodal few-shot learning models	MCAE with mined pairs	74.87 ± 1.86
	MTriplet with mined pairs	**85.49 ± 1.35**

5.5.1 Multimodal Five-Shot Speech-to-Image Matching

The main experiments of this chapter is shown in Table 5.1 which reports the multimodal five-shot 11-way speech-to-image matching accuracies. On this task, we consider the indirect few-shot models of Chapter 4, as well as the direct few-shot models of this chapter. The goal of this section is to establish whether the direct approach could eliminate the compounding of errors phenomenon that occurred in the indirect matching approach. In addition, this section intends to investigate whether learning a maping of spoken words and images to a single joint space, results in more accurate feature representations.

The top section of Table 5.1 shows the accuracies of the multimodal few-shot learning models of Chapters 3 and 4 on the indirect two-step matching approach. The bottom section reports the accuracies of the models considered in this chapter on the direct matching approach. From these results we see the direct few-shot models outperform the indirect transfer learned and unsupervised few-shot models, as well as the indirect model which is a combination of the unsupervised and transfer learning approaches (*Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE*). Since the direct models are also a combination of unsupervised and transfer learning, we conclude that unsupervised and transfer learning can be complementary and work quite well when combined in a direct multimodal few-shot manner.

From Table 5.1 we see that from all the models and approaches considered (in the entire thesis), the MTriplet is our best performing model. The MTriplet achieves an accuracy of 85.20% which outperforms even the MCAE (our second-best model) by a margin of roughly 10%. From this we conclude that the MTriplet produces the most general modality-invariant feature representations per class. To test this, Table 5.2 considers the per-digit multimodal recall scores for the models considered in Table 5.1.

The MTriplet achieves the highest per-digit recall scores for ten of the eleven classes by mostly achieving recall scores that are even significantly higher than that of its closest

Stellenbosch University https://scholar.sun.ac.za

Table 5.2: The per-digit recall scores of the MCAE and MTriplet on a five-shot 11-way speech-to-image matching task.

		Actual speech query digit class										
	Model	“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Recall(%)	Classifier (Table 4.4, row 1)	73.92	57.38	63.09	60.60	65.13	64.23	62.41	44.94	49.56	51.46	63.87
	Unsupervised CAE (Table 4.4, row 3)	78.37	37.87	51.27	44.14	41.20	47.74	59.20	45.11	41.78	59.78	65.22
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE	82.59	48.50	54.47	53.24	49.74	57.10	67.49	54.66	48.22	**65.68**	70.97
	MCAE	96.01	81.09	76.38	61.92	74.64	87.52	85.07	58.91	75.00	52.75	74.82
	MTriplet	**96.75**	**88.74**	**87.18**	**87.47**	**87.19**	**96.75**	**88.11**	**74.54**	**86.34**	65.58	**82.33**

competitor (the MCAE). This means that for the majority of digit classes, the MTriplet predicts the biggest fraction of word queries from a certain class to belong to its correct matching image. Logically this makes sense since the MTriplet objective function aims to distinguish between cross-modal representations of the same class and cross-modal representations from different classes. Similarly, the MCAE’s objective function aims to produce similar representations for same class cross-modal inputs. However, we can attribute their underperformance to the MTriplet to the idea that they should also retain enough information in the representations in order to produce a within-modality pair instance from the representation of an input. In retrospection, this would logically lead to within-modality nuisance information in the representations.

From Table 5.2, we also see that the MTriplet has significantly lower per-digit recall scores for the classes “eight” and “oh” than for the other classes. Specifically, the class “oh” is the only class in which the MTriplet does not achieve the highest recall score. To investigate this phenomenon, we consider the confusion matrix produced by the MTriplet in Table 5.3.

Just considering the class “oh” from these results, we see the MTriplet mostly confuses a speech query of an “oh” to be a *9*. Since an “oh” and a “nine” does not sound acoustically similar, we hypothesise that it might be that the MTriplet confuses images of a *0* and a *9* since they can look visually similar. However, we see that the MTriplet does not make the same mistake of predicting queries of a “zero” to be of a *9*. This means that for the class “zero” the MTriplet did not confuse images of a *0* and a *9*. A similar trend is seen in the MCAE’s confusion matrix in Table A.6, but the MCAE confuses speech queries of the class “oh” to belong to an *8*.

To further investigate this, in Table 5.4 we look at the confusion matrices for some classes produced by the models in Table 5.1 and Table 5.2. This table specifically considers

Stellenbosch University https://scholar.sun.ac.za

Table 5.3: The confusion matrix produced by the MTriplet with mined pairs on the five-shot 11-way speech-to-image matching task.

	Actual speech digit class										
	"one"	"two"	"three"	"four"	"five"	"six"	"seven"	"eight"	"nine"	"oh"	"zero"
1	8587	22	32	142	17	5	167	27	174	307	23
2	10	8142	770	26	0	16	195	203	23	56	553
3	3	319	8042	27	758	7	172	433	10	48	40
4	73	28	33	8025	56	80	154	437	106	486	46
5	3	2	119	113	7646	43	87	343	25	546	159
6	29	39	2	226	63	8683	4	171	3	63	574
7	7	438	118	32	17	8	8238	35	193	229	166
8	16	57	69	35	61	107	54	6634	664	380	39
9	130	7	35	365	103	0	172	600	7771	1069	43
0	17	121	5	184	54	26	107	17	31	6066	7657
Total	8875	9175	9225	9175	8775	8975	9350	8900	9000	9250	9300

the class "oh" and "zero" to investigate the phenomenon seen in the MTriplet, as well as the class "five" which was considered in Chapter 4. Considering the class "five", we see that the direct multimodal few-shot learning models confuses a query of a "five" to belong to a *3* (or any of the other nine classes) much less than what we see in the indirect

Table 5.4: Some of the confusion matrix classes produced by the direct and indirect multimodal few-shot learning models on the five-shot 11-way speech-to-image matching task. In order for the confusion matrices produced by the direct few-shot models (trained on five different batch sizes and five different seeds) to be comparable to those of the indirect models (trained on one batch size with five different seeds), we scale the direct models' scores down in this table. *The transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE.

			Predicted image class									
			1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Actual speech digit class	"five"	Indirect classifier	88	13	154	25	1143	96	23	132	43	38
		Indirect unsupervised CAE	117	17	258	74	723	132	62	127	120	125
		Indirect combination model*	99	19	241	59	873	123	49	121	93	78
		MCAE	40	2	263	24	1310	18	36	24	30	8
		MTriplet	3	0	152	11	1529	13	3	12	21	11
	"oh"	Indirect classifier	109	80	62	102	98	152	67	116	112	952
		Indirect unsupervised CAE	23	66	70	88	115	154	95	49	84	1106
		Indirect combination model*	21	39	51	66	106	162	64	45	81	1215
		MCAE	87	62	16	164	118	55	64	193	115	976
		MTriplet	61	11	10	97	109	13	46	76	214	1213
	"zero"	Indirect classifier	64	87	38	60	44	124	80	82	93	1188
		Indirect unsupervised CAE	22	60	54	37	112	159	81	30	92	1213
		Indirect combination model*	21	60	51	44	87	138	53	18	68	1320
		MCAE	4	171	10	94	11	60	70	10	38	1392
		MTriplet	4	111	8	9	32	115	33	8	9	1531

Stellenbosch University https://scholar.sun.ac.za

Table 5.5: Multimodal speech-to-image matching using the direct models and the direct approach vs. the indirect approach with a one-shot support set.

	Model	
	MCAE	MTriplet
Direct approach	**74.87 ± 1.86**	**85.49 ± 1.35**
Indirect approach	66.37 ± 2.62	76.18 ± 1.48

models. This trend holds for the class “zero” as well. However, the trend does not hold for the class “oh”. We see that the MCAE and MTriplet confuses the class “oh” just as much as the indirect models. In fact, the indirect combination model (*Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE*) confuses the class “oh” a bit less than the MTriplet and much less than the MCAE. This phenomenon in the class “oh” is investigated further in Chapter 5.5.2, but first we gain some further insights into the performance of the direct models.

5.5.1.1 Direct vs. Indirect Matching

The indirect approach to do the multimodal speech-to-image matching task requires two unimodal comparisons contrary to the one direct comparison required by the direct approach. In order to investigate the effect of using one vs. two comparisons, in Table 5.5 we separate the speech and vision subnetworks of the direct models so that we can use the respective speech and vision subnetworks as though they are separate networks and then apply these unimodal networks in an indirect (Chapter 3 and Chapter 4) matching approach.

Table 5.5 shows that using just one direct comparison in this multimodal matching task leads to higher matching accuracies than using two unimodal comparisons. From this, we conclude that the direct models perform better since there is no compounding of errors with just one comparison than with two unimodal comparisons as we saw in Chapter 4.

5.5.1.2 The Effect of the Hard Within-Modality Pairs

We train the MTriplet and the MCAE on mined cross-modal pairs, as well as within-modality speech or image positive or negative pairs mined with the hard restrictions set out in Chapter 5.3.3.2 and Chapter 5.3.3.3. Similarly, each of the combination speech CAEs in Chapter 4 were trained on the same hard speech positive pairs used to train the MCAE. However, the corresponding combination vision CAEs were not trained on the same hard image positive pairs used to train the MCAE (Chapter 5.3.3.2). Therefore, we ask which part of the direct few-shot models’ mined pairs attributed to the performance boost. Perhaps it is the speech-image pairs that forces the model to learn modality invariant

Stellenbosch University https://scholar.sun.ac.za

Table 5.6: Using hard within-modality positive image pairs on the vision CAE from Chapter 4.4.4. Both the "easier" and hard within-modality positive pairs are mined from representations extracted from a classifier.

Model	11-way accuracy (%)	
	One-shot	Five-shot
CAE (Table 4.13, row 3)	**48.66** ± **1.14**	55.59 ± 0.71
CAE with hard positive pairs	48.27 ± 1.08	**56.15** ± **0.50**

representations for the same class. Or perhaps it is the hard mined within-modality pairs.

To test whether the cross-modal pairs or within-modality positive pairs contributes most to the MCAE's performance, we consider the indirect multimodal few-shot CAE trained on pairs mined from classifier representations in Chapter 4.4.4. The training pairs of this multimodal CAE's vision network were not mined using hard restrictions. We compare this model in Table 5.6 to an indirect multimodal CAE with a vision network trained on the hard positive pairs mined from classifier representations.

Table 5.6 shows the hard positives pairs makes an insignificantly small difference to the one- and five-shot matching accuracies. This means a multimodal CAE trained on the same hard within-modality pairs as the MCAE does not improve the multimodal CAE in Chapter 4.4.4 which is trained on pairs mined from classifier representations. Although training on the combination of hard within-modality pairs together with the speech-image pairs could be the reason for the performance boost, we conclude that the speech-image pairs is the main contributor.

5.5.2 Unimodal Five-shot Classification Tasks

In order to investigate the performance contribution of the various parts of the direct multimodal five-shot learning models, we disconnect the direct models' speech and vision subnetworks. These separate networks are then tasked on unimodal five-shot classification tasks and compared to the unimodal five-shot classification accuracies reported in Chapter 4. The goal of this section is not to find the best unimodal results, but to gain insights into the performance of the multimodal models discussed in the previous subsection.

Table 5.7 shows the unimodal five-shot 11-way speech classification results for the speech networks of the direct and indirect multimodal few-shot learning models. Although not by far, it is surprising to see that the speech network of the indirect combined model (*Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE RNN*) outperforms the speech networks of both the MTriplet and MCAE since the MTriplet and MCAE outperforms the combined model on the multimodal matching task. In order to gain a bit more insight as to why this is happening, in Table 5.8 we consider the per-digit

Stellenbosch University https://scholar.sun.ac.za

Table 5.7: The unimodal five-shot 11-way speech classification task performed on the speech networks of the direct and indirect multimodal five-shot learning models.

Model		five-shot 11-way accuracy (%)
Indirect multimodal few-shot learning models	Transfer learned classifier RNN (from Table 4.14 row 1)	95.40 ± 0.50
	Unsupervised CAE RNN (from Table 4.14 row 2)	95.14 ± 0.80
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE RNN (from Table 4.14 row 4)	**97.91 ± 0.37**
Direct multimodal few-shot learning models	MCAE speech RNN	95.65 ± 1.09
	MTriplet speech RNN	97.25 ± 0.65

unimodal recall scores for the results reported in Table 5.7.

From Table 5.8 we see that for the per-digit speech recall scores, the speech network of the combined model achieves the highest accuracies for five of the eleven classes, while the classifier achieves the highest accuracies for three of the classes. The MTriplet only achieves the highest accuracies for two of the classes and the MCAE only for one of the classes. However, the MTriplet consistently achieves competitive results to those achieved by the classifier and combined speech network. A possible explanation might be that the speech networks of the indirect few-shot models captures within-modality information in their representations which helps them to outperform the MTriplet and MCAE by a small margin. In order to make logical conclusions from what this means with respect to the

Table 5.8: The per-digit recall scores of the speech networks of the MCAE and MTriplet on a five-shot 11-way speech classification task.

		Actual speech query digit class										
	Model	“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Recall(%)	Classifier (Table 4.8, row 1)	94.02	**99.13**	97.17	95.79	98.25	**99.95**	**99.24**	97.37	98.22	86.13	97.16
	Unsupervised CAE (Table 4.8, row 3)	95.16	88.86	96.51	96.12	95.38	98.58	98.05	96.66	98.28	88.29	95.26
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE RNN	**99.57**	97.72	**97.78**	**98.69**	98.37	98.31	98.70	**98.08**	**98.50**	95.30	97.10
	MCAE speech RNN	98.37	94.15	94.98	95.16	96.17	96.02	97.51	90.67	96.20	**95.85**	97.05
	MTriplet speech RNN	98.02	94.74	96.94	97.92	**98.41**	98.94	98.12	95.05	98.31	94.88	**98.46**

Stellenbosch University https://scholar.sun.ac.za

Table 5.9: The unimodal five-shot 10-way image classification task performed on the vision networks of the direct and indirect multimodal five-shot learning models.

Model		10-way five-shot accuracy (%)
Indirect multimodal few-shot learning models	Transfer learned classifier CNN (from Table 4.15 row 1)	82.90 ± 1.12
	Unsupervised CAE CNN (from Table 4.15 row 2)	77.62 ± 0.69
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE CNN (from Table 4.15 row 4)	81.60 ± 1.52
Direct multimodal few-shot learning models	MCAE vision CNN	90.06 ± 0.93
	MTriplet vision CNN	**90.46 ± 0.61**

multimodal results, we first have to consider the unimodal five-shot image classification scores of the corresponding vision networks.

Table 5.9 shows the unimodal five-shot 10-way image classification results achieved by the vision networks of the direct and indirect multimodal few-shot learning models. The trend seen in the image classification scores are similar to the trend seen in the multimodal speech-to-image matching accuracies in Table 5.1. However, there is one exception: the vision network of the MTriplet does not outperform the vision network of the MCAE by such a large margin than what is seen in the direct multimodal matching accuracy scores. To see what happens in these two vision networks, Table 5.10 considers the per-digit unimodal recall scores for the results reported in Table 5.9.

Table 5.10: The per-digit recall scores of the vision networks of the MCAE and MTriplet on a five-shot 10-way image classification task.

		Actual image query digit class									
	Model	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Recall(%)	Classifier (Table 4.10, row 1)	97.35	77.85	85.05	82.75	86.70	84.20	83.35	62.30	77.95	91.50
	Unsupervised CAE (Table 4.10, row 3)	98.75	68.00	79.50	68.05	69.10	75.85	82.70	70.30	70.20	93.70
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE CNN	99.05	75.25	81.45	73.70	73.70	82.90	83.70	77.30	73.75	95.20
	MCAE vision CNN	**99.15**	**89.33**	**92.57**	77.63	88.37	**95.81**	**91.05**	**80.56**	**88.71**	**97.43**
	MTriplet vision CNN	99.00	**89.33**	91.20	**86.26**	**93.17**	94.90	89.53	78.94	86.47	95.83

Stellenbosch University https://scholar.sun.ac.za

From the results in Table 5.10, we see that the MCAE achieves the highest recall scores for seven of the classes, while the MTriplet achieves the highest recall scores for two of the classes. For the query class “two”, the MCAE and MTriplet both achieve the highest recall score. Since the MCAE and MTriplet can classify image digits more accurately than the indirect models, we conclude that the direct models use the information provided by the word signals to better distinguish between the image classes. Overall, the MCAE achieves the highest recall scores for most of the digit classes with the MTriplet achieving competitive results to those of the MCAE. However, since the MCAE has significantly lower per-digit recall scores for the class *4* and *5* than those of the MTriplet, it results in the MCAE being a bit less accurate than the MTriplet on the unimodal five-shot image classification task.

Considering these speech and image classification scores together, we notice that although the MTriplet's speech network is not as specialised as some of the unimodal speech networks in the indirect models, it still performs competitively. This small price we pay is worth it since modelling the speech and image classes into a joint embedding space helps to better distinguish between the image digit classes and to find directly comparable speech and image representations. Therefore, we conclude that the MTriplet performs the best on the multimodal speech-to-image matching task since (1) its performance is consistently good over both modalities, and (2) it produces similar representations for speech and image instances of the same class.

The MCAE's vision network just falls short of the MTriplet's on the image classification task, but its speech network is outperformed by both the speech networks of the MTriplet and the indirect combined multimodal model. This will be investigated further in Chapter 5.5.4. We conclude that the MCAE performs second best on the multimodal speech-to-image matching task since (1) their vision network performs really well and its speech network performs competitively enough to the best performing speech networks, and (2) it is able to find similar speech and image representations for the same class.

In Chapter 4 we noticed that the unimodal results of the indirect multimodal few-shot learning models are consistently lower than their multimodal results. However, from the speech results in Table 5.7 and Table 5.8, the vision results in Table 5.9 and Table 5.10, as well as the multimodal results in Table 5.1 and Table 5.2 of the direct multimodal few-shot learning models, we see that although the multimodal results might still be a bit lower than the unimodal results, it is much less than what we observed from the indirect models.

Lastly, we recall the phenomenon we saw in the MTriplet and MCAE in Chapter 5.5.1 where these direct models achieved lower per-digit recall scores for the class “oh” than some of the indirect models. Specifically for the MTriplet, we saw that it confuses the class “oh” to be that of a *9*. Initially we hypothesised that since the words “oh” and “nine” are not acoustically similar, the network might not learn how to distinguish between the visual instances of a *0* and *9* since they can (in a subjective view) be visually very similar.

Stellenbosch University https://scholar.sun.ac.za

However, we concluded that since the MTriplet does not confuse queries of the class "zero" to be a *9*, the phenomenon in the class "oh" cannot be attributed to the visual instances of a *9* and *0* being too similar.

From the image recall scores for the class *0* in Table 5.10, we see that the MTriplet's vision network performs quite well. Its complete confusion matrix in Table E.4 shows that an image query of a *0* is rarely confused to be of a *9*. Turning to the speech recall score of the class "oh", we see that the MTriplet's speech network again performs quite well and its confusion matrix in Table C.3 shows that a speech query of an "oh" is hardly ever confused to belong to the class "nine". From this we conclude that although the MTriplet could find similar within-modality representations for images of *0*'s and for the words "oh", it could not find sufficiently similar cross-modal representations for the words of an "oh" and images of a *0*.

We see a similar phenomenon in the MCAE, where its speech network finds similar representations for the words "oh" (Table 5.8) and its vision network finds similar representations for the images of a *0* (Table 5.10), but the representations of an "oh" is not similar enough to the representations of a *0*. Since some of the unsupervised cross-modal pairs are incorrect, in the next subsection we investigate whether these incorrect pairs causes this confusion in the MTriplet and MCAE.

5.5.3 Is it possible to improve the MCAE and MTriplet?

We train the MCAE and MTriplet on mined unsupervised cross-modal pairs which means some of these cross-modal pairs might be incorrect since they are never checked. The goal of this subsection is to investigate whether this is the reason why the direct few-shot learning models cannot find similar enough cross-modal representations for some of the digit classes. To do this, we consider oracle results for the MTriplet and the MCAE.

The oracle results in Table 5.11 are idealised experiments showing the performance of the MCAE and MTriplet trained only on ground truth speech-image pairs and ground truth positive or negative within-modality pairs. We use the actual data labels to obtain

Table 5.11: Multimodal five-shot 11-way speech-to-image matching accuracies using the direct approach with the MTriplet and the MCAE trained on mined cross-modal pairs, as well as their oracle results.

Model	five-shot 11-way accuracy (%)
MCAE with mined pairs (Table 5.1, row 4)	74.87 ± 1.86
Oracle MCAE	93.64 ± 1.61
MTriplet with mined pairs (Table 5.1, row 5)	85.49 ± 1.35
Oracle MTriplet	99.10 ± 0.14

Stellenbosch University https://scholar.sun.ac.za

Table 5.12: The per-digit recall scores of the MCAE and MTriplet trained with mined pairs on a five-shot 11-way speech-to-image matching task, as well as their oracle results.

		Actual speech query digit class										
	Model	“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Recall(%)	MCAE	96.01	81.09	76.38	61.92	74.64	87.52	85.07	58.91	75.00	52.75	74.82
	Oracle MCAE	96.81	89.71	96.13	96.59	92.55	98.99	95.12	93.44	93.24	80.75	96.96
	MTriplet	96.75	88.74	87.18	87.47	87.19	96.75	88.11	74.54	86.34	65.58	82.33
	Oracle MTriplet	99.53	99.07	99.49	99.07	98.37	99.73	99.54	99.62	99.77	98.65	98.26

ground truth within-modality and cross-modal pairs. Since the few speech-image pairs from a multimodal support set are not the only data labels we use, it is important to note that the oracle models are not multimodal few-shot learning models.

From Table 5.11 we see that the oracle results achieves significantly higher accuracy scores than their multimodal few-shot learning counterparts. This leads us to believe that an improvement in the accuracy of the mined pairs could lead to an even bigger performance boost for the direct models. To gain more insight into whether the oracle models finds more general cross-modal representations for the same class, we consider the per-digit recall scores in Table 5.12.

The recall scores shows that the oracle models significantly improves the recall scores of each class. Specifically for the class “oh” which emerged as a problem class for the direct few-shot models, we see that for both the oracle MCAE and MTriplet, the recall scores for the class “oh” is significantly higher. Therefore, we conclude that the incorrect cross-modal pairs are to blame for the direct models’ low recall scores for the class “oh”. If we could improve the accuracy of the cross-modal pairs, we could find significantly better recall scores for all of the digit classes.

5.5.4 Speaker Invariance of the Speech Networks from the Direct Multimodal Models

Although we do not need the word representations in the direct matching approach to necessarily be speaker invariant, it is a good criterion to see whether the word representations only retains class information and filters out nuisance information like speaker identity. If a representation contains speaker information, the words “zero” and “oh” said by the same speaker might have more similar representations than the representations of the word “oh” said by two different speakers. We evaluate the word representations of the MTriplet and MCAE on the same hard five-shot 11-way speech classification tasks used in Chapter 4.4.3 to test the speaker invariance of the word representations produced by the indirect few-shot models’ speech networks.

This hard five-shot speech classification task uses episodes containing a single word

Stellenbosch University https://scholar.sun.ac.za

Table 5.13: The speaker invariance of the MCAE and MTriplet's speech networks on a five-shot 11-way speech classification task.

	Model	five-shot 11-way accuracy (%)
Indirect multimodal few-shot learning models	Classifier RNN (Table 4.12, row 1)	92.45 ± 0.83
	Unsupervised CAE RNN (Table 4.12, row 5)	87.75 ± 1.98
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE RNN	93.00 ± 1.08
Direct multimodal few-shot learning models	MCAE RNN	32.37 ± 5.22
	MTriplet RNN	**95.84 ± 1.25**

query and a five-shot speech support set. All the instances in the support set which are from the same class as the query, is sampled to be from a different speaker than the query. Each other instance in the support set which is from a different class as the query, is sampled to be from the same speaker as the query.

Table 5.13 shows the five-shot 11-way speech classification accuracies of the speech networks from the direct and indirect multimodal few-shot learning models. From these results, we see that the MTriplet produces the most speaker invariant representations. Once again this just proves that the MTriplet finds the most general representations per class. However, the MCAE achieves the lowest scores on this hard speech classification task, which is surprising since the MCAE performs fairly well on the classification and matching tasks throughout this chapter. This leads us to believe that the MCAE's speech network does not filter out enough nuisance information form its word representations. We therefore conclude that this is why the MCAE's speech network is outperformed by the speech networks of the MTriplet and the indirect combination model as seen in Chapter 5.5.2. In the same chapter we saw that the MCAE's vision network has the highest per-digit recall scores for most of the image classes. Therefore, we might be able to obtain a direct multimodal few-shot learning model that outperforms even the MTriplet if we are able to improve the MCAE's speech network.

5.6 Chapter Summary

In this chapter we explained the direct approach to do multimodal few-shot speech-to-image matching. For this direct approach we require direct multimodal few-shot learning models. We proposed and implemented two new direct multimodal few-shot learning models: the MCAE and the MTriplet. The MCAE is a completely new model, whereas the MTriplet trained on unsupervised in-domain cross-modal pairs has never been considered for this

Stellenbosch University https://scholar.sun.ac.za

task.

These direct few-shot models outperformed the indirect few-shot models of Chapter 4 which includes the purely unsupervised and transfer learning indirect models, as well as an indirect few-shot model which is a combination of unsupervised and transfer learning. Surprisingly the speech network of the indirect combination model outperformed the speech networks of the MCAE and MTriplet. However, the vision networks of this indirect combination model and the rest of the indirect models, struggled to find similar representations for each image digit class. This resulted in the indirect models often confusing the digit classes. To overcome this, the direct few-shot models used the spoken word information to find similar representations for each image digit class. This led to the vision networks of the direct models outperforming the vision networks of the indirect models.

Furthermore, the direct models found directly comparable representations for words and images, i.e. similar representations for spoken words and images of the same class. We concluded that although the speech networks of the direct models performed a bit worse than the indirect combination model's, the small price we pay is worth it to find much better image representations, as well as directly comparable cross-modal representations. By using these cross-modal representations in the direct matching approach, we eliminated the compounding of errors that emerges from the two-step indirect matching approach in Chapter 4.

The MTriplet came out as the best model outperforming even the second best model, the MCAE, by a significant margin. This indicates an objective function focussing on the similarity and dissimilarity of cross-modal input representations based on their class, works best. Upon further investigation into the MTriplet, we used the oracle experiments to show that a further improvement of its cross-modal pairs could lead to even better performance. Future work will focus on finding a better unsupervised mining process, as well as building more multimodal datasets to investigate the performance of these direct models on more difficult multimodal tasks.

Stellenbosch University https://scholar.sun.ac.za

Chapter 6

Summary and Conclusions

This thesis considered direct and indirect multimodal few-shot learning models to perform multimodal speech-to-image matching. In this multimodal matching task a model is given a few paired speech-image examples which it should use to match a given speech query to a matching image in a test set. We specifically considered both transfer and unsupervised learning: we compared the two methodologies within an indirect matching approach and then combined the two methodologies in a direct approach where speech and images are mapped to a single shared space.

6.1 Indirect Multimodal Few-Shot Learning

In Chapter 3, we re-implemented the indirect few-shot learning models proposed by Eloff et al. [1]. An indirect multimodal few-shot learning model consists of a speech and a vision network where each network aims to find similar representations for within-modality inputs of the same class. To do the speech-to-image matching task, an indirect model uses its speech network to do speech-speech comparisons to find a given speech query's closest speech instance in a multimodal support set of paired speech-image examples. Then the indirect model uses its vision network to do image-image comparisons to find the closest image instance in a matching set of images for the closest spoken instance's paired image.

Eloff et al. [1] specifically considered an indirect multimodal few-shot classifier and an indirect multimodal few-shot Siamese model. These indirect few-shot models consists of corresponding unimodal networks, e.g. the indirect multimodal classifier consists of a speech classifier and a vision classifier. Both networks in a multimodal model are trained on labelled background data not containing any of the few-shot digit classes seen at test time. Therefore, transfer learning is used to train these unimodal models since they are trained on out-of-domain data and then applied to do a task on unseen in-domain classes.

As in [1], we compared these transfer learned models to a baseline which uses speech-speech and image-image comparisons on raw speech and image inputs for the indirect matching approach. We repeated the experiments of [1] in order to improve the experimen-

Stellenbosch University https://scholar.sun.ac.za

tal setup and to ensure that all the models in this thesis would be comparable. As a side effect, we fixed an error in the validation setup of [1]. This resulted in a more reproducible setup which samples fixed episodes beforehand instead of sampling an episode on demand. After improving the experimental setup and evaluating the indirect classifier and Siamese few-shot models against the baseline, the classifier was identified as the best model.

In Chapter 4, we made our first novel contributions: indirect multimodal few-shot learning models that uses unsupervised learning. The idea behind using unsupervised learning is that before teaching an agent like a household robot new concepts from a few speech-image pairs per concept, it would be exposed to unlabelled in-domain data of these concept classes from its environment. Specifically, we considered three unsupervised objectives: the autoencoder (AE), correspondence autoencoder (CAE) and AE-CAE. These unsupervised models consists of an unsupervised speech network and a corresponding unsupervised vision network which are trained on unlabelled in-domain data. This means that during training the models see unlabelled instances of the few-shot digits seen at test time. Since the CAEs and AE-CAEs requires training pairs, we find unsupervised image pairs using cosine distance and unsupervised spoken word pairs using speaker information and dynamic time warping (DTW). We compare these unsupervised indirect few-shot models to the transfer learned classifier of Chapter 3, as well as transfer learned variants of the unsupervised CAE and AE-CAE. Similarly to the unsupervised indirect models, their transfer learned variants consist of a transfer learned speech network and a transfer learned vision network. The speech and vision transfer learned CAEs are trained on ground truth pairs from the background data which is obtained by using the actual data labels.

The motivation behind unsupervised and transfer learning are quite different. Transfer learning can be thought of as a way to re-use existing knowledge to make sense of unseen classes. In contrast, unsupervised learning involves an approach which learns from unlabelled observed data within the domain in which it will be used. From our comparison of the proposed indirect models, it became clear that transfer learning consistently outperformed unsupervised learning. To determine whether it is at all possible to obtain an unsupervised approach that could outperform the transfer learning approaches, we performed oracle experiments on the unsupervised models. From the oracle experiments we saw that by improving the pairs used to train the unsupervised models, we can find some unsupervised scheme that outperforms the pure transfer learning models.

As a preliminary investigation to determine whether the use of transfer learning can improve the unsupervised pairs, we used the transfer learned classifiers to extract representations which can then be used to obtain training pairs from the unlabelled in-domain data. We used these pairs to train unsupervised speech and vision CAEs to construct a new unsupervised indirect multimodal CAE. Additionally, we combined the unsupervised and transfer learning methodologies by pretraining a speech CAE and a vision CAE on ground truth pairs from the labelled background data before switching

Stellenbosch University https://scholar.sun.ac.za

to training these networks on the unsupervised pairs mined using the transfer learned classifiers. Unfortunately, after evaluating these indirect combination models we found it to just fall short of the transfer learned indirect classifier. However, these combination models showed improved performance over the pure unsupervised approach.

To gain further insights into the indirect multimodal few-shot models' performance, we also evaluate their speech and vision networks on unimodal speech or image classification tasks. In a unimodal classification task a model is given a support set containing a few unimodal labelled examples which it uses to predict the label of a given query. Specifically, by using the unimodal network, the given query is compared to each one of these labelled examples in the support set so that the query is classified according to the label of the its closest example.

From the unimodal and multimodal results we came to two conclusions. Firstly, since the unimodal classification results of the separate speech and vision networks are consistently higher than the multimodal speech-to-image matching results, there is a compounding of errors due to the two unimodal comparisons across the multimodal support set. Secondly, the vision networks could not find similar representations for images of the same digit classes. This resulted in the indirect multimodal few-shot models not being able to clearly distinguish between the image digit classes.

6.2 Direct Multimodal Few-Shot Learning

In Chapter 5, we combined the unsupervised and transfer learning approaches to obtain direct multimodal few-shot learning models in an effort to find more distinctive image representations for each digit class and to eliminate the compounding of errors in the indirect approach. These direct models attempts to learn a single multimodal space in which speech and image representations can be compared directly. Specifically, these models aim to find similar representations for speech and image instances of the same class. We then use these direct few-shot models to do the speech-to-image matching task in a direct approach: to match a given speech query to its matching image, the speech and image representations are compared directly in a joint space.

We proposed and implemented two direct multimodal few-shot learning models, the multimodal correspondence autoencoder (MCAE) and the multimodal triplet network (MTriplet). As a further contribution, the MCAE is an entirely new model. The MTriplet is based on previous models, but the MTriplet has not yet been trained on unsupervised mined cross-modal pairs and used for few-shot matching. For training, both the MCAE and MTriplet requires speech-image pairs, as well as positive (MCAE) or negative (MTriplet) speech-speech and image-image pairs. Within the mining procedure, we use the transfer learned classifiers to extract representations for speech and image instances from the unlabelled in-domain data. To mine training speech-image pairs, we use these

Stellenbosch University https://scholar.sun.ac.za

representations and the multimodal support set to pair up unlabelled in-domain speech and image instances. We also use these speech and image representations together with some hard restrictions to obtain image-image and speech-speech positive and negative pairs. None of these pairs are checked to be correct. Therefore the direct models learn a multimodal space from only the few ground truth speech-image pairs in the support set.

The results of our direct few-shot models showed that they outperform all the indirect few-shot models with the MTriplet achieving the highest multimodal matching accuracy scores. In order to further investigate the direct few-shot models, we performed oracle experiments for both direct models and we disconnected the speech and vision subnetworks of these direct models to test these subnetworks in isolation on unimodal classification tasks. From these extensive experiments we concluded that the direct models outperform the indirect models since they learn similar representations for speech and image inputs of the same digit class. These representations can be directly compared in the direct matching approach and therefore avoid the compounding of errors that emerged from the indirect approach.

6.3 Contributions

In order to emphasise the contributions of this thesis, we briefly state the aspects of the thesis we were the first to consider. The unsupervised unimodal AE, CAE and AE-CAE speech and vision models proposed in Chapter 4 has never been considered to construct indirect multimodal few-shot learning models. Similarly, we were also the first to consider transfer learned variants of this unsupervised indirect multimodal few-shot CAE and AE-CAE. Our study, to our knowledge, has also been the only one that combined unsupervised and transfer learning to obtain indirect few-shot models for the indirect matching approach. We are the first to consider the MCAE and MTriplet in a direct few-shot learning approach which combines the unsupervised and transfer learning methodologies to learn similar representations for speech and image inputs of the same class.

6.4 Recommendations for Future Work

Future work should look into finding more accurate methods of mining unsupervised cross-modal pairs, as well as speech-speech and image-image positive (MCAE) and negative (MTriplet) pairs. From our experiments in Chapter 5 we could already conclude that an improvement in the cross-modal and within-modality training pairs, will improve the direct few-shot MTriplet and MCAE. We could explore combining the CAE and triplet losses since Last et al. [85] has also found (on a different task than ours) that the CAE and triplet losses can be complementary.

Stellenbosch University https://scholar.sun.ac.za

Another avenue to pursue might be to investigate how these models can be used in the field of natural language processing. This will probably involve extending direct few-shot learning to consistently master new classes from a few speech-image pairs per new class as the model encounters these classes in its environment. To do this, meta-learning could be considered. Concretely, Eloff [82] performed preliminary studies into direct multimodal few-shot learning by considering a network similar to the MTriplet but using transfer learning in a meta-learning approach. The meta-learning approach is very different from the direct models proposed here. The two approaches could be compared in future work. This would require us to ensure that the results from [82] are reproducible (potentially incorporating some of the changes we had to make, as outlined in Chapter 3) and then applying the models to the same datasets within the same training regime.

Unlabelled multimodal datasets that consists of data in one modality and corresponding data from another modality, are scarce. Therefore, future work should investigate the collection of more multimodal datasets. Besides needing more multimodal datasets, in order to clearly investigate the feasibility of multimodal few-shot learning, we also need more realistic datasets which are applicable to real practical settings. To a large extent, MNIST is considered a toy problem which means future work should look into using natural images for multimodal few-shot learning.

6.5 Overall Summary and Conclusions

This thesis demonstrated two approaches for multimodal few-shot learning: a direct and an indirect approach. These few-shot models were used to do a multimodal speech-to-image matching task in which speech queries have to be matched to matching images after seeing only a few ground truth speech-image pairs (the multimodal support set). The indirect models attempt to find similar representations for within-modality inputs of the same class, which are used in the indirect approach to do the multimodal matching task. We compared using the unsupervised and transfer learning methodologies for these indirect models. In contrast, we showed that the direct few-shot models which combined the unsupervised and transfer learning methodologies to find similar representations for speech and image inputs of the same class, are more accurate than the indirect models.

More specifically, we showed that the direct few-shot learning MTriplet was our best model since its objective function focusses on distinguishing between speech and image representations from the same class and speech and image representations from different classes. The MCAE's objective function also attempts to produce similar representations for speech and image inputs of the same class. However, the MCAE should also retain enough information in the representations to be able to produce other within-modality instances of the same class from these representations. Upon retrospection, this is not ideal since the aim of direct multimodal few-shot learning is to only learn which characteristics

Stellenbosch University https://scholar.sun.ac.za

inputs from the same class across different modalities have in common. Overall, these direct few-shot models shows promise for finding systems that do not require large amounts of labelled data while simultaneously being able to quickly link data from different modalities.

Stellenbosch University https://scholar.sun.ac.za

References

[1] R. Eloff, H. A. Engelbrecht, and H. Kamper, "Multimodal one-shot learning of speech and images," in *Proc. ICCASP*, 2019.

[2] H. Kamper, "Unsupervised neural and Bayesian models for zero-resource speech processing," Ph.D. dissertation, University of Edinburgh, UK, 2017.

[3] L. Fei-Fei, Fergus, and Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. ICCV*, 2003.

[4] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. PAMI*, vol. 28, 2006.

[5] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum, "One shot learning of simple visual concepts," *CogSci*, vol. 33, 2011.

[6] B. M. Lake, C.-y. Lee, J. R. Glass, and J. B. Tenenbaum, "One-shot learning of generative speech concepts," *CogSci*, vol. 36, 2014.

[7] G. Koch, "Siamese neural networks for one-shot image recognition," in *Proc. ICML*, 2015.

[8] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. NIPS*, 2016.

[9] P. Shyam, S. Gupta, and A. Dukkipati, "Attentive recurrent comparators," in *Proc. ICML*, 2017.

[10] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. NIPS*, 2017.

[11] A. Borovsky, J. L. Elman, and M. Kutas, "Once is enough: N400 indexes semantic integration of novel word meanings from a single exposure in context," *Lang Learn Dev*, vol. 8, 2012.

[12] I. Biederman, "Recognition-by-components: A theory of human image understanding." *Psych. Review*, vol. 94, 1987.

Stellenbosch University https://scholar.sun.ac.za

[13] G. A. Miller and P. M. Gildea, “How children learn words,” *SciAM*, vol. 257, 1987.

[14] R. L. Gómez and L. Gerken, “Infant artificial language learning and language acquisition,” *TiCS*, vol. 4, 2000.

[15] O. Räsänen and H. Rasilo, “A joint model of word segmentation and meaning acquisition through cross-situational learning.” *Psych. Review*, vol. 122, 2015.

[16] B. van Niekerk, L. Nortje, and H. Kamper, “Vector-quantized neural networks for acoustic unit discovery in the ZeroSpeech 2020 challenge,” *arXiv preprint arXiv:2005.09409*, 2020.

[17] C. Lee, T. J. O’Donnell, and J. Glass, “Unsupervised lexicon discovery from acoustic input,” *Trans. ACL*, vol. 3, 2015.

[18] H. Kamper, “Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models,” in *Proc. ICCASP*, 2019.

[19] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, 2009.

[20] S. Ruder, “Neural transfer learning for natural language processing,” Ph.D. dissertation, NUI Galway, Ireland, 2019.

[21] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, “Unsupervised neural network based feature extraction using weak top-down constraints,” in *Proc. ICCASP*, 2015.

[22] A. Parnami and M. Lee, “Few-shot keyword spotting with prototypical networks,” *arXiv:2007.14463*, 2020.

[23] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in *Proc. ICLR*, 2017.

[24] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, “Meta-learning with memory-augmented neural networks,” in *Proc. ICML*, 2016.

[25] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” *arXiv preprint arXiv:1703.03400*, 2017.

[26] W. Thomason and R. A. Knepper, “Recognizing unfamiliar gestures for human-robot interaction through zero-shot learning,” in *Proc. ISER*, 2017.

[27] D. Wu, F. Zhu, and L. Shao, “One shot learning gesture recognition from RGBD images,” in *Proc IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2012.

Stellenbosch University https://scholar.sun.ac.za

[28] J. Wan, Q. Ruan, W. Li, and S. Deng, "One-shot learning gesture recognition from RGB-D data using bag of features," *J. Mach. Learn. Res.*, vol. 14, 2013.

[29] T. Stafylakis and G. Tzimiropoulos, "Zero-Shot keyword spotting for visual speech recognition in-the-wild," in *Proc. ECCV*, 2018.

[30] M. R. Walter, Y. Friedman, M. Antone, and S. Teller, "One-shot visual appearance learning for mobile manipulation," *IJRR*, vol. 31, 2012.

[31] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," *arXiv:1709.04905*, 2017.

[32] B. M. Lake, R. R. Salakhutdinov, and J. Tenenbaum, "One-shot learning by inverting a compositional causal process," in *Proc. NIPS*, 2013.

[33] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, 2015.

[34] B. Lake, R. Salakhutdinov, and J. Tenenbaum, "Concept learning as motor program induction: A large-scale empirical study," *CSSAC*, vol. 34, 2012.

[35] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," *arXiv preprint arXiv:2003.08505*, 2020.

[36] R. Salakhutdinov, J. Tenenbaum, and A. Torralba, "One-Shot learning with a hierarchical nonparametric Bayesian model," *JMLR Workshop Conf Proc*, vol. 27, 2012.

[37] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *Proc. ICLR*, 2018.

[38] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: A good embedding is all you need?" *arXiv preprint arXiv: 2003.11539*, 2020.

[39] J. Lyons, "Python speech features," 2013.

[40] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans Acoust Speech Signal Process*, vol. 28, 1980.

[41] R. G. Leonard and G. R. Doddington, "TIDIGITS LDC93S10," Philadelphia: Linguistic Data Consortium, 1993.

Stellenbosch University https://scholar.sun.ac.za

[42] M. A. Pitt, K. Johnson, E. Hume, S. Kiesling, and W. Raymond, "The Buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability," *Speech Commun.*, vol. 45, 2005.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, 1998.

[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[45] Y. Gao and D. Glowacka, "Deep Gate Recurrent Neural Network," in *Proc. ACML*, 2016.

[46] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU," *J. Artif. Intell. Soft Comput. Res.*, vol. 9, 2019.

[47] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *JMLR*, vol. 3, 2002.

[48] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "Siamese" time delay neural network," in *Proc. NIPS*, 1994.

[49] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. CVPR*, vol. 1, 2005.

[50] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. CVPR*, 2014.

[51] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *Pattern Recognit. Image Anal.*, vol. 5524, 2009.

[52] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, 2015.

[53] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. SIMBAD*, 2015.

[54] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv:1703.07737*, 2017.

[55] K. M. Hermann and P. Blunsom, "Multilingual distributed representations without word alignment," in *Proc. ICLR*, 2014.

[56] D. Chicco, P. Sadowski, and P. Baldi, "Deep autoencoder neural networks for gene ontology annotation predictions," in *Proc. ACM*, 2014.

Stellenbosch University https://scholar.sun.ac.za

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[58] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems."

[59] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, 2012.

[60] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, 2012.

[61] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. ICML*, 2014.

[62] H. Kamper, Y. Matusevych, and S. Goldwater, "Improved acoustic word embeddings for zero-resource languages using multilingual transfer," *arXiv preprint arXiv:2006.02295*, 2020.

[63] D. Harwath, A. Torralba, and J. Glass, "Unsupervised learning of spoken language with visual context," in *Proc. NIPS*, 2016.

[64] L. Nortje and H. Kamper, "Unsupervised vs. transfer learning for multimodal one-shot matching of speech and images," *arXiv preprint arXiv:2008.06258*, 2020.

[65] Y.-A. Chung, C.-C. Wu, C.-H. Shen, and H.-y. Lee, "Unsupervised learning of audio segment representations using sequence-to-sequence recurrent neural networks," in *Proc. Interspeech*, 2016.

[66] Y.-H. Wang, H.-y. Lee, and L.-s. Lee, "Segmental audio Word2Vec: Representing utterances as sequences of vectors with applications in spoken term detection," in *Proc. ICCASP*, 2018.

Stellenbosch University https://scholar.sun.ac.za

[67] N. Holzenberger, M. Du, J. Karadayi, R. Riad, and E. Dupoux, "Learning word embeddings: Unsupervised methods for fixed-size representations of variable-length speech segments," in *Proc. Interspeech*, 2018.

[68] Y. Chung, W. Weng, S. Tong, and J. Glass, "Unsupervised cross-modal alignment of speech and text embedding spaces," in *Proc. NIPS*, 2018.

[69] J. Koutník, J. Schmidhuber, and F. Gomez, "Evolving deep unsupervised convolutional networks for vision-based reinforcement learning," in *Proc. GECCO*, 2014.

[70] G. E. Hinton, "Reducing the dimensionality of data with neural metworks," *Science*, 2006.

[71] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: feature learning by inpainting," in *Proc. CVPR*, 2016.

[72] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural metworks," in *Proc. NIPS*, 2012.

[73] K. Barnard, P. Duygulu, D. Forsyth, and N. de Freitas, "Matching words and pictures," *Mach Learn*, vol. 3, 2003.

[74] R. Socher and L. Fei-Fei, "Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora," in *Proc. CVPR*, 2010.

[75] D. Harwath and J. Glass, "Deep multimodal semantic embeddings for speech and images," in *Proc. ASRU*, 2015.

[76] D. Harwath, W.-N. Hsu, and J. Glass, "Learning hierarchical discrete linguistic units from visually-grounded speech," *arXiv:1911.09602*, 2020.

[77] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proc. ACM*, 2014.

[78] C. Silberer and M. Lapata, "Learning grounded meaning representations with autoencoders," in *Proc. ACL*, 2014.

[79] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. ICML*, 2011.

[80] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Proc. NIPS*, 2013.

[81] J. Weston, S. Bengio, and N. Usunier, "Large scale image annotation: learning to rank with joint word-image embeddings," *Mach Learn*, vol. 81, 2010.

Stellenbosch University https://scholar.sun.ac.za

[82] R. Eloff, “Multimodal one-shot learning of speech and images,” Thesis, Stellenbosch University, Stellenbosch, 2020.

[83] K. Leidal, D. Harwath, and J. Glass, “Learning modality-invariant representations for speech and images,” in *Proc. ASRU*, 2017.

[84] D. Harwath, A. Recasens, D. Suris, G. Chuang, A. Torralba, and J. Glass, “Jointly discovering visual objects and spoken words from raw sensory input,” in *Proc. ECCV*, 2018.

[85] P.-J. Last, H. A. Engelbrecht, and H. Kamper, “Unsupervised feature learning for speech using correspondence and Siamese networks,” *IEEE Signal Process. Lett.*, vol. 27, 2020.

Stellenbosch University https://scholar.sun.ac.za

Appendix A

The confusion matrices of the multimodal models

In Chapter 4 we consider multimodal few-shot learning models which perform a multimodal speech-to-image matching task using an indirect approach. Specifically, we consider these models on an indirect five-shot 11-way speech-to-image matching task. Table A.1, Table A.2, Table A.3 and Table A.4 shows the confusion matrices for some of these tasks. Table A.5 shows the confusion matrix for the multimodal CAE trained on oracle pairs and applied to the indirect five-shot 11-way speech-to-image matching task.

The multimodal few-shot learning models considered in Chapter 5 performs the multimodal speech-to-image matching task using a direct approach. Table A.6 shows the confusion matrix for one of these models, the MCAE, on a direct five-shot 11-way multimodal speech-to-image matching task.

Table A.1: The confusion matrix produced by the multimodal transfer learned CAE on the five-shot 11-way speech-to-image matching task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted image class	*1*	1329	158	71	71	146	63	138	128	159	78	21
	2	63	838	140	76	25	82	126	110	35	62	86
	3	23	182	1018	28	208	27	59	167	71	50	32
	4	32	53	44	922	73	196	78	104	240	94	43
	5	30	25	200	54	736	82	46	95	109	138	99
	6	49	74	31	154	107	1019	42	125	57	157	144
	7	79	202	92	67	71	15	1111	62	159	51	48
	8	84	127	140	119	104	136	71	767	201	59	42
	9	68	44	83	255	133	28	154	166	711	90	65
	0	18	132	26	89	152	147	45	56	58	1071	1280
	Total	1775	1835	1845	1835	1755	1795	1870	1780	1800	1850	1860

Stellenbosch University https://scholar.sun.ac.za

Table A.2: The confusion matrix produced by the multimodal unsupervised CAE on the five-shot 11-way speech-to-image matching task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted image class	*1*	1391	260	128	113	117	66	123	145	92	23	22
	2	108	695	176	127	17	87	114	143	68	66	60
	3	32	136	946	47	258	48	49	141	72	70	54
	4	1	108	41	810	74	206	92	87	306	88	37
	5	37	27	228	53	723	127	50	145	140	115	112
	6	29	86	46	155	132	857	21	102	68	154	159
	7	61	188	54	58	62	52	1107	52	130	95	81
	8	55	112	116	91	127	113	59	803	126	49	30
	9	28	92	70	313	120	95	177	118	752	84	92
	0	3	131	40	68	125	144	78	44	46	1106	1213
	Total	1775	1835	1845	1835	1755	1795	1870	1780	1800	1850	1860

Table A.3: The confusion matrix produced by the multimodal unsupervised CAE with transfer learned classifier pairs on the five-shot 11-way speech-to-image matching task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted image class	*1*	1425	221	150	88	119	77	89	111	79	23	15
	2	84	828	137	86	25	92	125	122	51	34	65
	3	36	121	991	40	212	33	59	133	54	46	49
	4	23	92	44	891	67	196	78	96	308	70	42
	5	51	20	194	47	819	103	36	151	118	94	66
	6	15	77	44	180	138	897	25	118	82	153	127
	7	52	170	59	87	39	44	1143	53	153	72	67
	8	66	130	144	83	137	113	51	807	131	50	26
	9	21	77	58	285	108	90	149	150	774	74	75
	0	2	99	24	48	91	150	115	39	50	1234	1328
	Total	1775	1835	1845	1835	1755	1795	1870	1780	1800	1850	1860

Table A.4: The confusion matrix produced by the multimodal transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE on the five-shot 11-way speech-to-image matching task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted image class	*1*	1466	175	77	44	99	70	84	63	65	21	21
	2	75	890	139	68	19	82	118	90	42	39	60
	3	32	138	1005	29	241	33	46	112	63	51	51
	4	22	81	37	977	59	153	70	87	311	66	44
	5	26	25	265	32	873	106	27	117	86	106	87
	6	19	53	47	145	123	1025	16	109	60	162	138
	7	57	208	70	61	49	21	1262	49	129	64	53
	8	53	103	115	84	121	122	41	973	133	45	18
	9	24	84	67	357	93	51	136	140	868	81	68
	0	1	78	23	38	78	132	70	40	43	1215	1320
	Total	1775	1835	1845	1835	1755	1795	1870	1780	1800	1850	1860

Stellenbosch University https://scholar.sun.ac.za

Table A.5: The confusion matrix produced by the multimodal CAE with oracle pairs on the five-shot 11-way speech-to-image matching task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted image class	*1*	1715	18	14	19	13	12	10	5	23	5	3
	2	5	1634	7	2	0	0	26	21	1	3	17
	3	0	46	1776	0	60	1	7	4	6	0	0
	4	5	9	3	1669	13	10	9	7	213	14	1
	5	3	0	21	1	1550	15	3	6	26	13	3
	6	1	12	1	13	37	1733	2	35	0	13	14
	7	5	51	8	3	27	0	1786	8	22	8	6
	8	1	37	9	8	23	12	1	1675	35	10	3
	9	40	2	5	113	29	1	19	12	1463	3	2
	0	0	26	1	7	3	11	7	7	11	1781	1811
	Total	1775	1835	1845	1835	1755	1795	1870	1780	1800	1850	1860

Table A.6: The confusion matrix produced by the MCAE with mined pairs on the five-shot 11-way speech-to-image matching task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	“five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted image class	*1*	8521	26	84	689	201	34	15	465	685	434	23
	2	10	7440	1118	98	8	47	285	343	52	311	853
	3	23	454	7046	82	1317	30	311	247	319	83	51
	4	59	44	26	5681	120	382	178	571	192	819	469
	5	63	13	303	511	6550	148	130	256	271	590	54
	6	5	94	25	496	89	7855	63	1293	28	275	302
	7	14	459	409	89	179	99	7954	97	239	318	349
	8	49	236	74	93	121	317	55	5243	377	966	51
	9	125	49	116	229	147	28	238	324	6750	575	190
	0	6	360	24	1207	43	35	121	61	87	4879	6958
	Total	8875	9175	9225	9175	8775	8975	9350	8900	9000	9250	9300

Stellenbosch University https://scholar.sun.ac.za

Appendix B

The per-digit recall scores of the multimodal models

Chapter 3 and Chapter 4 considers using unsupervised or transfer learning or a combination of these two methodologies for indirect multimodal few-shot learning. These models are used on an indirect approach to do a multimodal five-shot 11-way speech-to-image matching task. The per-digit multimodal recall scores for these models are reported in Table B.1.

Table B.1: The per-digit recall scores of the multimodal combination models on a five-shot 11-way speech-to-image matching task.

		Actual speech query digit class										
	Model	“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Recall(%)	Classifier	73.92	**57.38**	**63.09**	**60.60**	**65.13**	**64.23**	62.41	44.94	**49.56**	51.46	63.87
	Unsupervised CAE	78.37	37.87	51.27	44.14	41.20	47.74	59.20	45.11	41.78	59.78	65.22
	Unsupervised CAE with transfer learned classifier pairs	80.28	45.12	53.71	48.56	46.67	49.97	61.12	45.34	43.00	**66.70**	**71.40**
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE	**82.59**	48.50	54.47	53.24	49.74	57.10	**67.49**	**54.66**	48.22	65.86	70.97
	Oracle pairs CAE	96.62	89.05	96.26	90.95	88.32	96.55	95.51	94.10	81.28	96.27	97.37

Stellenbosch University https://scholar.sun.ac.za

Appendix C

The confusion matrices of the speech models

Chapter 4 considers various indirect multimodal few-shot learning models which consist of separate speech and vision networks. Table C.1 and Table C.2 shows the confusion matrices of these speech networks on unimodal five-shot 11-way speech classification tasks.

Chapter 5 considers direct multimodal few-shot learning models. After disconnecting the direct models' speech and vision subnetworks, we use the speech subnetworks to do unimodal five-shot 11-way speech classification tasks. Table C.3 shows the confusion matrix for the MTriplet's speech network on a unimodal five-shot 11-way speech classification task.

Table C.1: The confusion matrix produced by the speech transfer learned CAE RNN on the five-shot 11-way speech classification task.

		Actual speech digit class										
		"one"	"two"	"three"	"four"	five"	"six"	"seven"	"eight"	"nine"	"oh"	"zero"
Predicted speech class	"one"	1681	0	3	1	7	0	0	25	102	66	1
	"two"	0	1633	20	8	0	1	20	26	0	10	129
	"three"	0	53	1744	4	2	1	3	50	6	7	2
	"four"	0	5	5	1727	25	0	2	1	0	75	7
	"five"	3	0	6	28	1691	4	9	7	17	83	0
	"six"	0	3	1	1	0	1797	24	15	0	0	6
	"seven"	0	13	5	5	1	17	1750	0	1	4	35
	"eight"	6	16	13	1	0	7	2	1658	3	48	0
	"nine"	123	2	4	1	11	0	0	25	1664	34	12
	"oh"	27	6	0	38	38	0	0	17	6	1476	2
	"zero"	0	109	4	16	0	8	35	1	1	7	1601
	Total	1840	1840	1805	1830	1775	1835	1845	1825	1800	1810	1795

Stellenbosch University https://scholar.sun.ac.za

Table C.2: The confusion matrix produced by the unsupervised speech CAE RNN on the five-shot 11-way speech classification task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted speech class	“one”	1751	0	2	10	5	0	0	4	26	16	0
	“two”	0	1635	33	1	0	1	12	15	0	2	43
	“three”	0	49	1742	1	4	0	2	2	2	0	1
	“four”	11	5	0	1759	25	3	1	3	0	47	9
	“five”	1	0	0	27	1693	0	8	1	1	71	0
	“six”	0	9	0	5	0	1809	10	11	0	0	7
	“seven”	0	23	3	2	6	12	1809	1	0	19	20
	“eight”	5	34	11	1	0	4	0	1764	0	37	0
	“nine”	67	0	5	0	21	0	0	13	1769	16	5
	“oh”	5	15	2	16	21	0	3	11	1	1598	0
	“zero”	0	70	7	8	0	6	0	0	1	4	1710
	Total	1840	1840	1805	1830	1775	1835	1845	1825	1800	1810	1795

Table C.3: The confusion matrix produced by the speech network of the MTriplet with mined pairs on the five-shot 11-way speech classification task.

		Actual speech digit class										
		“one”	“two”	“three”	“four”	“five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Predicted speech class	“one”	9018	2	13	5	2	0	0	7	96	44	6
	“two”	0	8716	157	3	0	28	62	132	0	6	90
	“three”	0	141	8749	1	28	3	13	109	0	4	1
	“four”	6	47	19	8960	22	0	14	0	1	119	1
	“five”	1	5	25	48	8734	17	37	33	18	167	1
	“six”	0	7	0	3	12	9078	0	42	0	1	0
	“seven”	0	85	12	5	4	8	9052	1	0	9	14
	“eight”	0	61	42	2	9	40	0	8673	11	24	0
	“nine”	139	1	3	12	36	0	0	108	8848	83	9
	“oh”	36	36	3	96	27	0	14	20	19	8587	16
	“zero”	0	99	2	15	1	1	33	0	7	6	8837
	Total	9200	9200	9025	9150	8875	9175	9225	9125	9000	9050	8975

Stellenbosch University https://scholar.sun.ac.za

Appendix D

The per-digit recall scores of the speech models

Chapter 3 and Chapter 4 considers using unsupervised or transfer learning or a combination of these two methodologies for indirect multimodal few-shot learning. The speech networks of these few-shot models are used on unimodal five-shot 11-way speech classification tasks. The per-digit speech recall scores for these tasks are reported in Table D.1.

Table D.1: The per-digit recall scores of the speech combination models on a five-shot 11-way speech classification task.

		Actual speech query digit class										
	Model	“one”	“two”	“three”	“four”	five”	“six”	“seven”	“eight”	“nine”	“oh”	“zero”
Recall(%)	Classifier RNN	94.02	**99.13**	97.17	95.79	98.25	**99.95**	**99.24**	97.37	98.22	86.13	**97.16**
	Unsupervised CAE RNN with cosine pairs	95.16	88.86	96.51	96.12	95.38	98.58	98.05	96.66	98.28	88.29	95.26
	Unsupervised CAE RNN with transfer learned classifier pairs	98.37	94.84	95.73	98.09	97.69	98.31	97.72	96.60	98.17	94.31	96.27
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE RNN	**99.57**	97.72	**97.78**	**98.69**	**98.37**	98.31	98.70	**98.08**	**98.50**	**95.30**	97.10
	Oracle pairs CAE RNN	99.77	96.14	98.28	99.56	98.65	99.84	99.08	99.18	99.44	98.23	99.39

Stellenbosch University https://scholar.sun.ac.za

Appendix E

The confusion matrices of the vision models

The indirect multimodal few-shot learning models in Chapter 3 and Chapter 4 consist of separate speech and vision networks. Table E.1, Table E.2 and Table E.3 show the confusion matrices of these vision networks on unimodal five-shot 10-way image classification tasks.

In Chapter 5 we consider direct multimodal few-shot learning models which consists of speech and vision subnetworks. These vision subnetworks are used to do unimodal five-shot 10-way image classification tasks. Table E.4 shows the confusion matrix for the MTriplet's vision network on a unimodal five-shot 10-way image classification task.

Table E.1: The confusion matrix produced by the vision classifier CNN on the five-shot 10-way image classification task.

		Actual image digit class									
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Predicted image class	*1*	1947	57	12	83	28	61	49	75	42	10
	2	13	1557	57	15	0	7	65	68	12	15
	3	6	93	1701	3	68	1	44	90	31	3
	4	13	12	0	1655	1	54	26	67	59	23
	5	0	7	85	7	1734	39	3	96	19	3
	6	4	28	0	62	74	1684	0	88	10	36
	7	7	106	56	31	7	3	1667	38	120	31
	8	7	88	62	60	52	67	24	1246	116	19
	9	0	4	16	57	10	13	98	138	1559	30
	0	3	48	11	27	26	71	24	94	32	1830
	Total	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000

Stellenbosch University https://scholar.sun.ac.za

Table E.2: The confusion matrix produced by the transfer learned vision CAE CNN on the five-shot 10-way image classification task.

		Actual image digit class									
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Predicted image class	*1*	1971	106	16	35	63	42	57	56	34	5
	2	8	1496	100	32	3	11	85	61	10	8
	3	5	64	1519	6	151	2	39	90	36	3
	4	3	17	5	1424	26	89	20	68	208	12
	5	2	12	152	15	1397	38	7	66	53	16
	6	0	38	13	133	95	1614	0	109	30	49
	7	3	157	63	28	9	2	1612	26	103	10
	8	6	66	87	70	91	56	28	1346	117	5
	9	0	11	33	225	46	10	98	137	1369	7
	0	2	33	12	32	119	136	54	41	40	1885
	Total	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000

Table E.3: The confusion matrix produced by the unsupervised vision CAE CNN on the five-shot 10-way image classification task.

		Actual image digit class									
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Predicted image class	*1*	1975	240	31	58	39	54	68	79	34	5
	2	12	1360	76	42	8	22	73	58	8	7
	3	2	72	1590	11	192	4	40	63	30	3
	4	0	51	8	1361	48	81	16	48	271	9
	5	5	7	129	17	1382	59	11	101	45	19
	6	0	33	24	112	76	1517	2	82	27	50
	7	3	141	36	45	13	3	1654	38	87	16
	8	1	54	51	39	112	73	14	1406	59	6
	9	1	19	35	293	65	38	63	95	1404	11
	0	1	23	20	22	65	149	59	30	35	1874
	Total	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000

Table E.4: The confusion matrix produced by the vision network of the MTriplet with mined pairs on the five-shot 10-way image classification task.

	Actual image digit class									
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
1	9900	21	11	18	6	97	25	76	34	11
2	35	8933	188	20	1	8	304	158	4	26
3	4	151	9120	16	150	0	142	358	80	6
4	3	65	14	8626	17	45	32	250	330	37
5	0	0	321	21	9317	145	5	236	90	107
6	36	46	11	313	198	9490	0	297	1	54
7	13	422	95	115	3	0	8953	92	440	83
8	9	127	208	258	201	85	43	7894	282	46
9	0	24	25	600	57	3	436	527	8647	47
0	0	211	7	13	50	127	60	112	92	9583
Total	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000

Stellenbosch University https://scholar.sun.ac.za

Appendix F

The per-digit recall scores of the vision models

The multimodal few-shot learning models in Chapter 3 and Chapter 4 use unsupervised or transfer learning or a combination of these two methodologies. The vision networks are used to do unimodal five-shot 10-way image classification tasks where Table F.1 shows the per-digit image recall scores of these tasks.

Table F.1: The per-digit recall scores of the vision combination models on a five-shot 10-way image classification task.

		Actual image query digit class									
	Model	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
Recall(%)	Classifier CNN	97.35	**77.85**	**85.05**	**82.75**	**86.70**	**84.20**	83.35	62.30	**77.95**	91.50
	Unsupervised CAE RNN	98.75	68.00	79.50	68.05	69.10	75.85	82.70	70.30	70.20	93.70
	Unsupervised CAE CNN with transfer learned classifier pairs	**99.05**	75.25	81.45	73.70	73.70	82.90	**83.70**	**77.30**	73.75	95.20
	Transfer learning + unsupervised fine-tuning with transfer learned classifier pairs CAE CNN	98.95	72.25	80.50	73.70	71.90	77.45	81.35	74.25	70.55	**95.60**
	Oracle pairs CAE CNN	100.00	97.05	98.75	96.75	97.70	96.75	99.05	96.80	93.00	99.85